TVP5160EVM User's Guide

Digital Video Department

Contents

Figures

Tables

1 Functional Description

The TVP5160EVM refers to both the TVP5160 board and the ADV7311 board when they are connected together. Both boards share a common interface via a 120-pin connector. This interface provides all data, clocks, I^2C communication, and 5-V power to each board.

The ADV7311 is a professional grade, 12-bit, 216-MHz, video encoder. This device minimizes potential artifacts caused by the re-encode process. When evaluating the TVP5160 decoder, it is highly recommended that the YPbPr component video outputs of the ADV7311 be used. This will ensure the video decoder within the display monitor will not be used.

1.1 Description Overview

The TVP5160EVM is powered by a single, 5-V, universal supply. I^2C communication is emulated using a PC parallel port configured for ECP (extended capability port) or bidirectional mode. The parallel port mode can be changed using the PC BIOS setup, available during the reboot process.

The TVP5160 video decoder converts the analog video input signal into digital component data. This digital data and the associated clocks from the video decoder are sent to the ADV7311 video encoder. The video encoder converts the digital data back into analog video. The analog video outputs provided by the ADV7311 encoder include CVBS, S-Video (SV), and YPbPr. These are all output simultaneously.

Control of the TVP5160EVM is provided by WinVCC4, a Windows-based application developed by Texas Instruments and provided free of charge. This application uses the parallel port to provide I^2C communication to the TVP5160EVM. WinVCC4 provides a graphics user interface (GUI) and a register level interface to program and vary the controls available within the TVP5160 decoder and the ADV7311 encoder.

2 Board Level Description

Figure 1 illustrates the various features available on the TVP5160EVM.

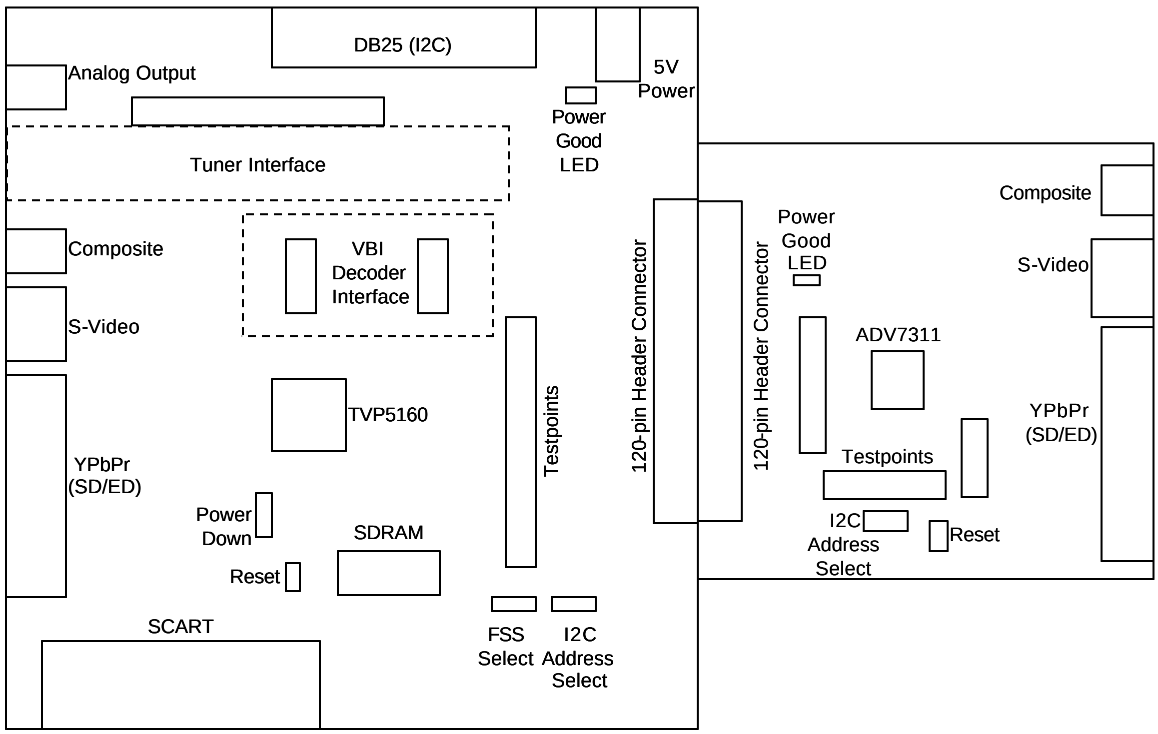


Figure 1. TVP5160EVM Block Diagram

2.1 Analog Inputs

The TVP5160EVM makes use of all the available inputs on the TVP5160 decoder. The following inputs are available for use:

- Composite (CVBS)
- S-Video
- YPbPr (SD/ED)
- SCART (CVBS and component RGB)
- Tuner interface (CVBS)
- VBI decoder interface (analog/digital RGB)

NOTE: The S-Video (Y/C) input is shared with the SCART (CVBS and R) inputs and must not be connected simultaneously.

Table 1 shows the pins used for the inputs described above.

Table 1. Analog Inputs and Pin Terminals

Input Type	TVP5160 Pin(s)
CVBS	VI_1
S-Video	VI_3(Y), VI_9(C)
YPbPr	VI_5(Y), VI_11(Pb), VI_8(Pr)
SCART	VI_3(CVBS), VI_9(R), VI_6(G), VI_12(B)
Tuner interface (CVBS)	VI_2
VBI decoder interface (analog RGB)	VI_7(R), VI_4(G), VI_10(B)

2.2 Anti-Aliasing Filters

Each analog input has an anti-alias filter installed by default to ensure the input to the TVP5160 decoder is of the best quality and does not alias. The filters are elliptical in design in order to minimize the cost and still provide the best roll-off. Details on the anti-aliasing filters, including frequency response and the group delay, are provided in a separate application note.

The filter on any analog input may be bypassed by removing the 0-? resistors currently installed (JP1-6, 9, 10) and shorting one of them across 1-2. Since the S-Video input is shared with the SCART (CVBS and R), these inputs share the same anti-aliasing filters.

2.3 Analog Output

The analog output from the TVP5160 decoder is made available on the TVP5160EVM. The output is buffered using a simple emitter follower and the output signal is automatically determined by the TVP5160 decoder. If a CVBS is selected as the input, then the analog output is the same looped-through CVBS output. If S-Video or YPbPr is the selected input, then only Y is output since it contains the sync information. Typically, the analog output is used for VCR record functions in some CE applications.

2.4 Tuner Interface

The tuner interface provides a method of connecting an analog TV tuner to the TVP5160EVM. The expected output from the tuner module to the TVP5160EVM is baseband composite video. Termination for the 75-? input source is already provided on the TVP5160EVM tuner input. The pinout of the tuner interface is provided in Table 2.

Table 2. Tuner Interface Pinout (H2)

Description	Pin Number
CVBS output (from tuner)	10
SDA (I^{2}C data)	29
SCL (I^{2}C address)	31
GND	1-9, 11-28, 30, 32-36
D5V	37-40

To supply the tuner module, D5V supply and GND pins are provided. The I^{2}C pins, SDA and SCL, are connected to the I^{2}C communications bus on the TVP5160EVM. Control of the tuner module is available within the TVP5160EVM software as discussed later.

2.5 VBI Decoder Interface

This interface provides a method of connecting an external VBI decoder module to the TVP5160EVM via the H3 and H4 headers. The VBI decoder may be a closed-caption decoder, teletext decoder, or any other VBI decoder currently available on the market that provides the same signals. The pinout of the VBI decoder interface is provided below.

Table 3. VBI Decoder Interface Pinout

Description	H3 Pin Number	H4 Pin Number
Tuner CVBS input (from tuner I/F)	1	-
CVBS input (from TVP5160EVM)	3	-
Analog RGB	11, 9, 7 (R, G, B)	-
HSYNC	4	-
VSYNC	6	-
FSS (for analog overlay)	10	-
Digital RGB	-	7, 9, 11 (DR, DG, DB)
SDA (I^2C data)	-	8
SCL (I^2C address)	-	10
FSO (for digital overlay)	-	12
GND	2, 5, 8, 12	5, 6
D5V	-	1-4

With this interface, there are two methods of overlaying the RGB character data output from the VBI decoder onto the input signal to the TVP5160 decoder:

- Analog RGB overlay
- Digital RGB overlay

Separate sets of the RGB inputs are provided depending on which overlay method is preferred. The analog RGB inputs are connected to the TVP5160 analog inputs as discussed in Section 2.1. The digital RGB inputs and FSO are connected to the digital R, G, and B inputs and FSO which are shared with the C bus, C6-9, respectively.

2.6 Test Points and Jumpers

Various test points are available on the TVP5160EVM for the user. This includes the various power supplies as well as a few GND test points. The primary test-point header is H6 and provides access to the video data, video clocks, I^2C, 5 V, and GND.

There are three jumpers available on the TVP5160 board that configure the power down mode, I^2C address select, and the FSS select. Each jumper is set by default in its preferred state for the TVP5160EVM. Next to each jumper on the TVP5160 board is the silkscreen that describes the two states of the jumper configuration.

If the I^2C address is changed on either the TVP5160 board or the ADV7311 board while the TVP5160EVM is powered up, then that device will not recognize the new I^2C address. The reset button on the TVP5160EVM must be pressed and WinVCC4 must be reconfigured for the new I^2C address.

2.7 SDRAM

On the TVP5160EVM, there are three SDRAM footprints: 50-pin, 54-pin, and 86-pin. These are provided to accommodate the user's choice of SDRAM that will be used in their final product. By default, the TVP5160EVM is provided with 64-Mbit SDRAM (54TSOPII).

If the SDRAM is changed, then the SDRAM configuration register (0x59) will also need to be updated with the correct memory size. This must be programmed before 3DYC or 3DNR is enabled.

2.8 Common Board Interface

The TVP5160EVM uses a 120-pin connector to connect the TVP5160 board to the ADV7311 board. This interface shares all common signals including the I^2C and the 5-V supply. The purpose is to modularize the TVP5160 board and allow users to interface it to a variety of other Texas Instruments products including triple video DACs, DVI transmitters, or to any other platform that shares the same interface.

This connector shares all digital video data (Y[9:0], C[9:0]), all video clocks (SCLK, VS, HS, GLCO, AVID, and FID), reset, I^2C, and 5-V as mentioned above.

2.9 Video Encoder Analog Outputs

The analog outputs of the ADV7311 board include CVBS, S-Video, and YPbPr. All of the outputs are available simultaneously. For evaluation purposes, it is recommended that the YPbPr component video outputs be used in order to bypass the video decoder internal to the display monitor.

3 System Level Description

The system block diagram illustrated in Figure 2 provides an example of how the TVP5160EVM may be used for evaluation. Typically, the analog input is a CVBS signal provided by a video source such as a pattern generator or a DVD player running a test DVD.

The TVP5160EVM itself is configured with the provided 5-V supply and the parallel port cable. The analog output is typically YPbPr to reduce the number of artifacts caused by backend processing or re-encoding. These outputs are then fed into a high-end or studio-quality display monitor such as a Sony Trinitron.

At the same time, the CVBS output from the encoder may also be fed into a video test measurement system such as the Tektronix VM700. This allows various tests to be run and also allows the user to analyze the video waveform or vectorscope.

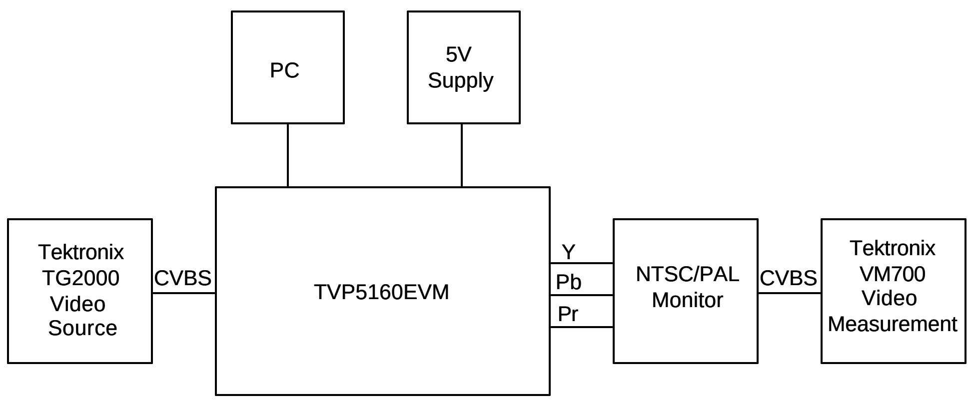


Figure 2. TVP5160EVM System Level Block Diagram

4 Required Hardware and Equipment

The following lists the required hardware and equipment necessary to use the TVP5160EVM:

- TVP5160EVM (provided)
- Universal 5-V power supply (provided)
- Parallel cable (provided)
- Windows-based PC
- 1 composite RCA video cable
- 1 YPbPr component RCA video cable
- Video source (pattern generator, DVD player, etc)
- TV or display monitor that supports YPbPr component video inputs

5 Hardware Setup

The following describes how to set up the hardware for the TVP5160EVM.

1. Connect the TVP5160EVM boards together using the 120-pin board connector on each board.
2. Connect a CVBS input to the TVP5160 board and a component cable to the YPbPr outputs of the ADV7311 board.

   NOTE: For evaluation it is recommended that the YPbPr component video outputs be used in order to bypass the internal video decoder of the TV or video display.
3. Connect the parallel port cable from the TVP5160EVM to the PC.

   NOTE: There are footprints for a dc jack and a DB25 connector on the ADV7311 board, but the default power and I^2C is provided by the TVP5160 board via the 120-pin connector, P2.
4. Connect the 5-V power supply to the dc jack on the TVP5160 board. A green LED on each board should now be lit.

6 Software Installation

WinVCC4 is a Windows application that uses the PC parallel port to emulate I^2C, providing access to each device on the I^2C bus. WinVCC4 makes use of CMD files, a text editable file that allows preset video setups to be programmed easily.

This feature allows the user to easily set multiple I^2C registers with the press of a button. WinVCC4 also has “Property Sheets” for the TVP5160 which allows the user to control the I^2C registers with a GUI.

All necessary software for the TVP5160EVM is provided on the enclosed CD. The following provides the steps required to install WinVCC4:

1. Insert the provided TVP5160EVM CD.
2. Install Port95NT.exe. This is the parallel port driver used by WinVCC4. This driver must be installed and the PC must be rebooted before WinVCC4 will operate correctly.
3. Install Setup.exe. Click *Next* at all prompts and click *Finish* to complete the installation process. This will install WinVCC4 onto the PC. No reboot is required.
4. Run WinVCC4.exe

NOTE: A shortcut to WinVCC4 should now be available on the desktop. WinVCC4 and additional TVP5160 related documentation can also be found at *Start->Programs-> TVP5160EVM Software*.

7 WinVCC4 Quick Start

The following describes the steps to take within WinVCC4 in order to get video out of the TVP5160EVM.

1. Once WinVCC4 is executed, the WinVCC4 Configuration screen appears. This dialog box is used to configure the I^2C bus. Next to VID_DEC, select the TVP5160 and ensure the I^2C address is set to 0xB8. This should match the I2C ADDR jumper on the TVP5160 board.

   NOTE: If WinVCC4 is running and the TVP5160 board I^2C address is changed, power must be cycled on the EVM.

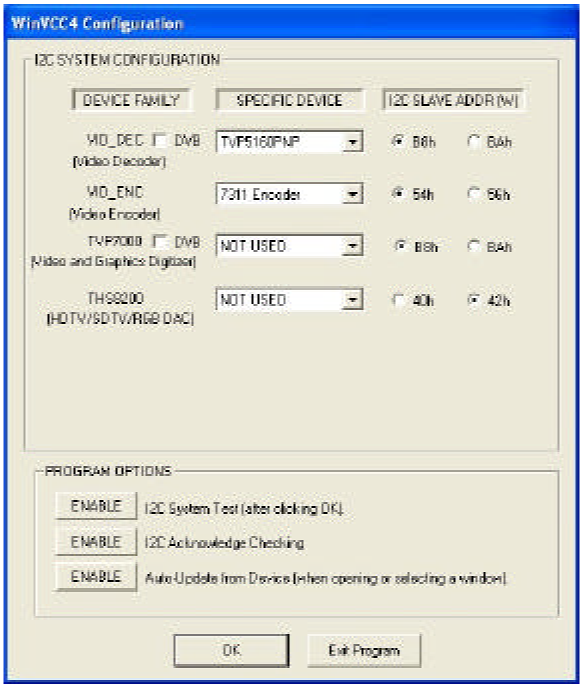


Figure 3. WinVCC4 – I^2C Configuration Screen

2. Next to VID_ENC, select the ADV7311 and ensure the I^2C address is set to 0x54. This should match the I2C ADDR jumper on the ADV7311 board.

3. Ensure that all other boxes are selected as Not Used and that all “Program Options” buttons are set to Enabled. Click OK.

4. If there are no I^2C communication issues, then the Real-Time Polling window will display next. If there are I^2C issues, an I2C Test Report box will display. Completely exit out of WinVCC4, double check the parallel port cable connections, cycle power on the TVP5160EVM and re-run WinVCC4.

5. At the Real-Time Polling window, ensure that VIDEO-STANDARD AUTO_SWITCH POLLING is enabled and click OK.

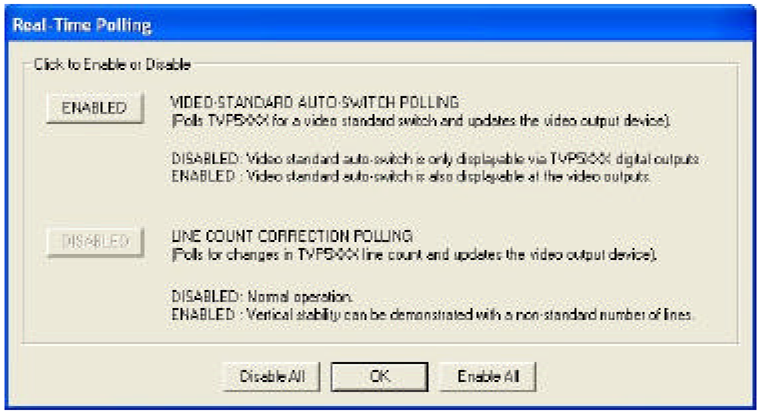


Figure 4. WinVCC4 – Real Time Polling Screen

6. Load the provided TVP5160EVM.CMD file into WinVCC4 by clicking on *Tools -> System Initialization -> Browse*. The CMD should be available by default as seen below.



Figure 5. WinVCC4 – Main Screen

7. Click the “TVP5160 (SD) + ADV7311” dataset in the window and then click the Program Dataset button to initialize the TVP5160EVM.

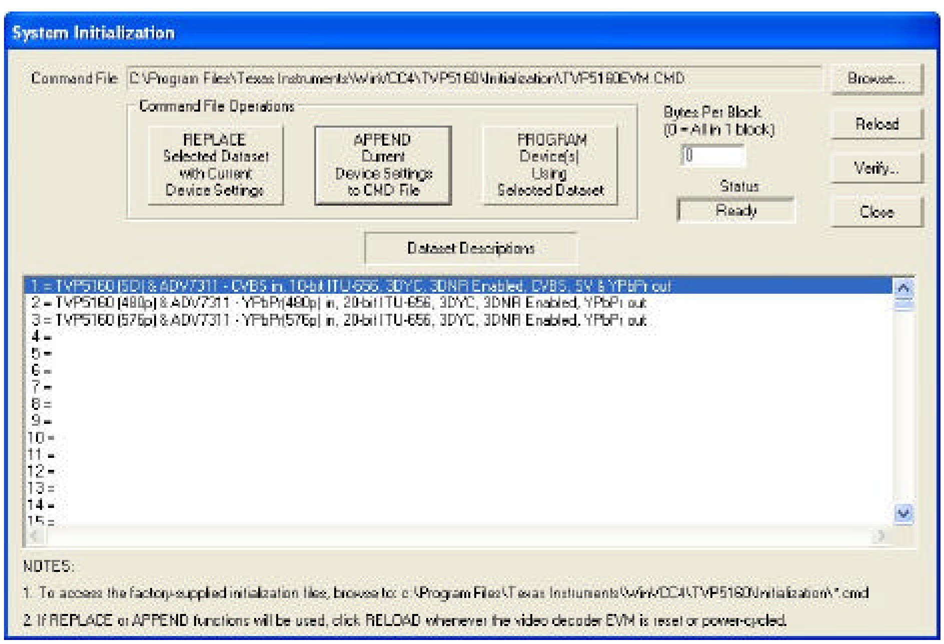


Figure 6. WinVCC4 – System Initialization

8. With a video source provided at the composite video input, video should be viewable on the TV/display monitor. All ADV7311 outputs are enabled simultaneously.

   NOTE: To ensure the TVP5160 is working properly, go to Video Status by clicking on *Edit -> Property Sheets -> TVP5160PNP -> Video Decoder Status* and check the H/V/C lock status and the video standard. This is only a check on the TVP5160 board and not the ADV7311 board or the TV/display monitor.

8 WinVCC4 in Depth

The following describes how to use WinVCC4 in depth. It discusses various features and screens which the user may encounter while evaluating the TVP5160EVM.

8.1 Starting WinVCC4

The Port95NT parallel port driver must be installed before using WinVCC4. WinVCC4 may be started by clicking on *Start->All Programs->TVP5160EVM Software->WinVCC4.*

If the dialog shown in the figure below is displayed, it means one of two things:

1. WinVCC4 did not run to completion the last time it ran. In this case, click *OK* to exit the program and restart WinVCC4.
2. There is more than one instance of WinVCC4 running at the same time. In this case, click *OK* to exit the program. Then, press CTRL-ALT-DELETE to bring up the *Task Manager*. Select and click *End Task* on all occurrences of WinVCC4 or WinVCC4 CONFIGURATION. Then restart WinVCC4.

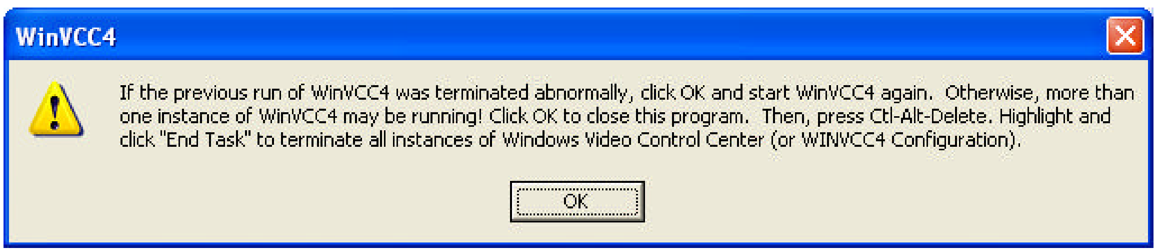


Figure 7. WinVCC4 Multiple Occurrences Error Message

8.2 WinVCC4 Configuration Dialog Box

The *WinVCC4 Configuration* dialog box, as seen in Figure 8, should now be visible. This dialog configures the I^{2}C bus on the TVP5160EVM. All settings from this dialog box are stored in the Windows registry and are restored the next time the program is started. After initial installation, VID_DEC™ will be set to TVP5160 and VID_ENC will be set to ADV7311.

The I^{2}C slave address for each device must match the I^{2}C slave address selected by jumpers on the TVP5160EVM. These jumpers are set by the factory to use 0xB8 for the video decoder and 0x54 for the video encoder.

It is also important to select the correct specific devices. The TVP5160 and ADV7311 must be selected for the TVP5160EVM.

All Program Options must be enabled. Disabling these options is only required if you are debugging a problem with the I^{2}C bus itself.

Clicking *OK* begins I^{2}C communication with the selected devices.

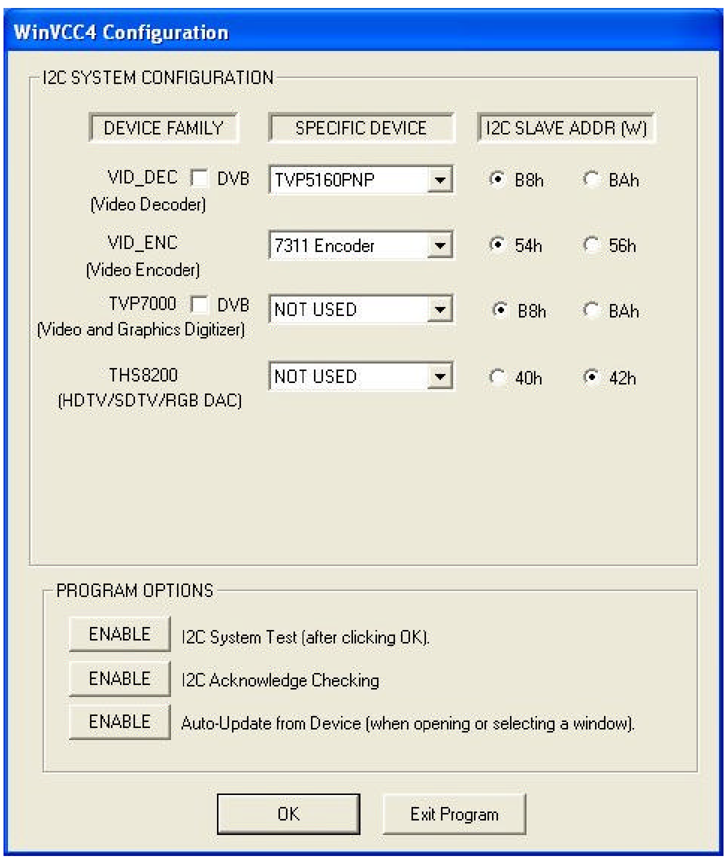


Figure 8. WinVCC4 I^{2}C Address Configuration

8.3 I²C System Test

The I²C system test of selected registers runs immediately after closing the *WinVCC4 Configuration* dialog box with *OK* (unless the I²C system test program options button was disabled).

If the I²C system test passes, then only a PASS message will appear. If the test failed, then a dialog box like Figure 9 will appear. See Section 9, *Troubleshooting*, for details on how to resolve this issue.

The I²C system test can be run at anytime by clicking *Run System* I2C *Test* in the *Tools* menu.

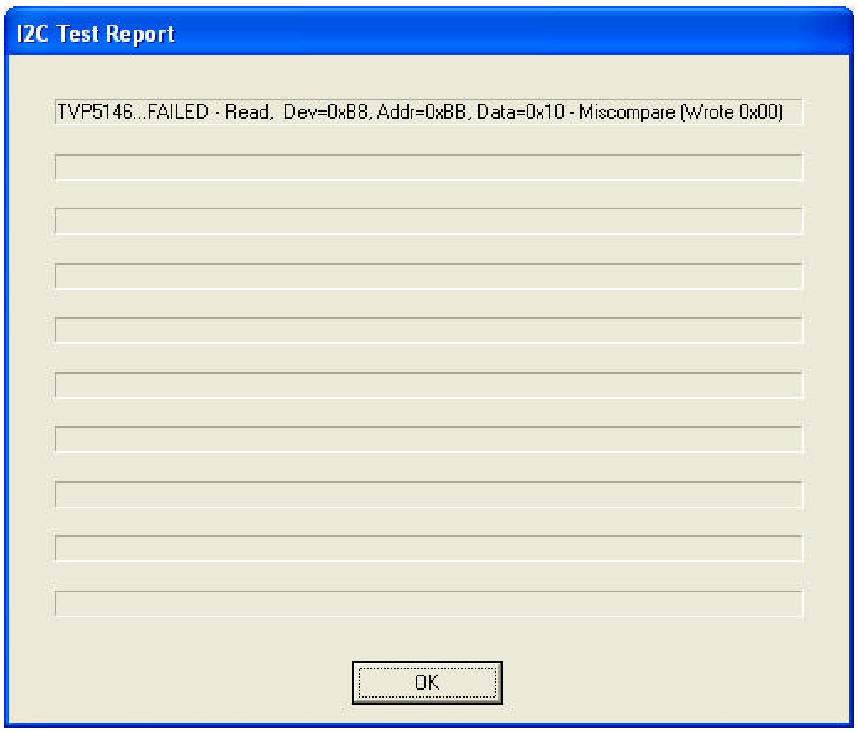


Figure 9. I²C System Failure

8.4 Real-Time Polling

Real-time polling provides polling functions that execute in the background continuously, when enabled via the *Real-Time Polling* dialog. There are two polling functions. The function that applies to the TVP5160 decoder is VIDEO–STANDARD AUTO–SWITCH POLLING.

When the TVP5160 decoder detects a change in the input video standard, it automatically switches to operation in the detected standard (which includes all necessary I²C register initialization) for proper decoding of the input video. The ADV7311 encoder does not have this feature. For this reason WinVCC4 must update the video encoder.

If the WinVCC4 autoswitch polling function is enabled, then the detected video standard status from the TVP5160 decoder is polled until a change in the input video standard (or in the TVP5160 sampling mode) is detected. When a change is detected, the video encoder is reprogrammed as needed for the detected standard. Using this feature, the video source can change its video standard and the system will display using the new standard without user intervention.

To enable autoswitch polling (recommended), the video-standard autoswitch polling function must be enabled in the *Real–Time Polling* dialog box as shown in Figure 10. The *Real–Time Polling* dialog can also be accessed once WinVCC4 is up and running by clicking *Real–Time Polling* in the *Tools* menu.

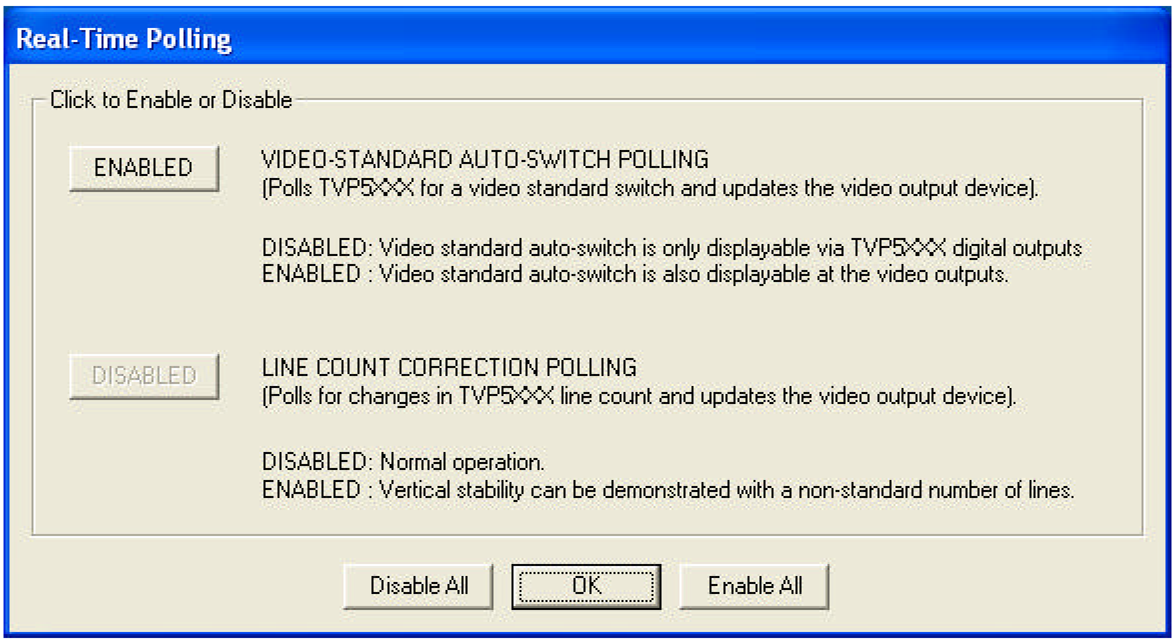


Figure 10. Real-Time Polling Dialog Box

8.5 Main Menu

After closing the *Real–Time Polling* dialog, the main menu is displayed as shown in Figure 11. The menus, which are used to operate WinVCC4, are *File*, *Edit*, *Tools*, *Window*, and *Help*. The *File* menu's only function is *Exit*, which terminates the program. The following table summarizes the main menu contents.



Figure 11. WinVCC4 – Main Screen

Table 4. Main Menu Summary

Menu	Contents
File	Exit
Edit	Register Map
	TVP5160PNP
	7311 Encoder Module Editor
	Generic I^{2}C Editor
	Property Sheets
	TVP5160PNP
	7311 Encoder Module Editor
Tools	System Initialization
	Real-time Polling
	TV Tuner Control (FQ12xx series only)
	Multiple-Byte I^{2}C Transfers
	Set I^{2}C Bit Rate
	Run System I^{2}C Test
	Run Continuous I^{2}C Test
	Read VBI FIFO
	Capture Live VBI Data
Window	Allows selection of the active window. Multiple windows can be open at the same time.
Help	Displays program version

8.5.1 System Initialization

Clicking *System Initialization* in the *Tools* menu displays the dialog shown below. This provides the means for initializing the video decoder and/or video encoder for a particular video mode. The details of the initialization are contained in the command file (with a CMD file extension).

The command file is loaded using the *Browse*… button. Once the command file is opened, a text list displays descriptions of the individual data sets contained within the command file.

Click once on the desired data set description to select it. Click the *Program Device(s) Using Selected Dataset* button to run the selected data set, which loads the devices via the I^2C bus. When the device initialization has completed, the status indicator reads *Ready*.

NOTE: If *Ready* does not display, then the devices are not initialized and the I^2C bus is not communicating. See Section 9, *Troubleshooting*, for possible solutions.

Click the *Close* button to close the dialog box. Each time the *System Initialization* dialog is closed, the initialization file pathname and the data set selection number are saved in the Windows™ registry to allow these settings to be retained for the next time *WinVCC4* runs.

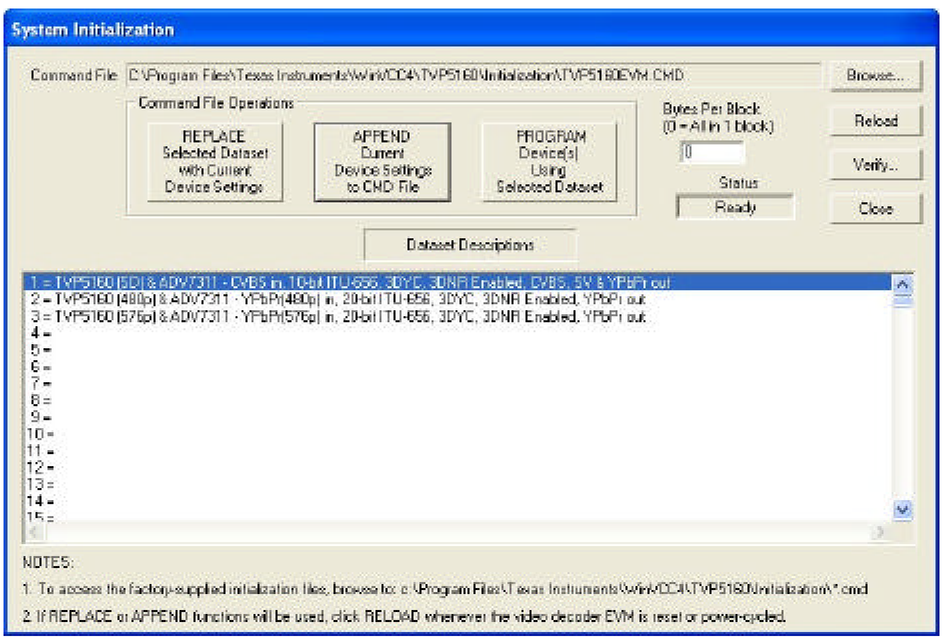


Figure 12. System Initialization

8.5.1.1 Adding a Custom Data Set

After you program the EVM via the *System Initialization* tool using the factory-supplied command file, and by using the *Property Sheets* tool, you can customize the device register settings to suit your needs. To save your custom settings:

1. Reopen the *System Initialization* dialog via the *Tools* menu.
2. Click the *Append Current Device Settings to Command File* button. A dialog requesting a description of the new data set appears.
3. Optionally, click the drop-down box and select one of the existing descriptions.
4. Modify the description text or type your own description.
5. Click *OK*. All nondefault register values from the decoder and encoder will be appended to the current command file as an additional data set.

Now, you can select your custom data set and send it with a press of the *Program*… button.

NOTE: The command file (.CMD) must be saved as plain text.

8.5.1.2 Command Files

The command file is a text file that can be generated using any common editor; however, it must be saved as plain text. Command files are especially useful for quickly switching between the various system configurations. These .CMD files are unrelated to the typical Windows™ .CMD files.

A default command file has been provided on the CD. This command file should contain most of the desired setups. This command file is located at:

c:\Program Files\Texas Instruments\WinVCC4\TVP5160\Initialization Files\Initialize.cmd

A command file can contain up to 250 data sets. A data set is a set of register settings to initialize the video decoder and/or video encoder for a particular video mode. Each data set includes a description that is displayed in one row of the dataset descriptions list. The register settings may be located in the command file itself and/or may be stored in separate include file(s) (with an .INC file extension) and be included into the command file using the INCLUDE statement.

8.5.1.3 Example Command File

An example of one data set within a command file is shown below. Each command file may contain individual *write to register* (WR_REG) commands.

```
BEGIN_DATASET                                            // Dataset 1

DATASET_NAME,"TVP5160 (SD) & ADV7311 - CVBS in, 10-bit ITU-656, 3DYC, 3DNR Enabled,
CVBS, SV & YPbPr out"

WR_REG,VID_DEC,0x01,0xEE,0x01      // Holds processor in reset
WR_REG,VID_DEC,0x01,0xEA,0xB0   // Enables fast processor mode
WR_REG,VID_DEC,0x01,0xE9,0x00
WR_REG,VID_DEC,0x01,0xE8,0x63
WR_REG,VID_DEC,0x01,0xE0,0x01
WR_REG,VID_DEC,0x01,0xEE,0x00      // Releases processor reset

// TVP5160 I2C Registers - SD
WR_REG,VID_DEC,0x01,0x04,0x3F // Auto Switch Mask
WR_REG,VID_DEC,0x01,0x06,0x40 // Pedestal off
WR_REG,VID_DEC,0x01,0x34,0x11 // Outputs Enabled
WR_REG,VID_DEC,0x01,0x35,0x2A // GPIO =0; GLCO, AVID and FID Enabled
WR_REG,VID_DEC,0x01,0x36,0xAF // HS and VS Enabled
WR_REG,VID_DEC,0x01,0x75,0x1A // F & V Bit Control
WR_REG,VID_DEC,0x01,0x7F,0x01 // Analog Output Enabled
WR_REG,VID_DEC,0x01,0x59,0x07 // Enabled SDRAM and set SDRAM size
WR_REG,VID_DEC,0x01,0x0D,0x84 // Chrominance Processing Ctrl 1 Reg – 3DYC, 3DNR Enabled

// ADV7311 I2C Registers - NTSC, all DACs Output
WR_REG,0x54,0x01,0x01,0x08         // CLK Align Enabled, SD Oversampled
WR_REG,0x54,0x01,0x40,0x10         // Selects NTSC, SSAF Luma Filter
WR_REG,0x54,0x01,0x42,0x41         // Enables DAC Outputs, PrPb SSAF
WR_REG,0x54,0x01,0x44,0x06         // RTC Enabled
WR_REG,0x54,0x01,0x48,0x14         // Enables Double Buffering, SD 10-bit, DNR disabled
WR_REG,0x54,0x01,0x7C,0x02         // Global 10-bit enable

END_DATASET
```

1. The comment indicator is the double-slash //.
2. The command file is not case-sensitive and ignores all white-space characters.
3. All numbers can be entered as hexadecimal (beginning with 0x) or as decimal.
4. Every data set in a command file begins with BEGIN_DATASET and ends with END_DATASET. The maximum number of datasets is 250.
5. The dataset text description is entered between double quotes using the DATASET_NAME command. The enclosed text can be up to 128 characters in length. This text appears in the *System Initialization* dialog when the command file is opened.
6. The INCLUDE command inserts the contents of an include file (with an .INC file extension) in-line in place of the INCLUDE command. Therefore, the include file should not contain the BEGIN_DATASET, END_DATASET, and DATASET_NAME commands.

   NOTE: All included files must be located in the same directory as the command (CMD) file.

7. The *write to register* command is written as follows:

   WR_REG, <DeviceFamily>, <Number of data bytes (N)>, <subaddress>, <Data1>,…, <DataN>

   or

   WR_REG, <Literal slave address>, <Number of data bytes (N)>, <subaddress>, <Data1>,…, <DataN>

   The valid device family mnemonics are:

   VID_DEC for the video decoders

   VID_ENC for the video encoders

   THS8200 for the THS8200 device

   WinVCC4 translates the device family mnemonic to the slave address that was selected on the *WinVCC4 Configuration* dialog upon program startup. This eliminates having to edit command files if the alternate slave address must be used.

   If the literal slave address method is used, then the slave address entered will be used directly. This method is normally used for programming the video encoder.

8. A delay may be inserted between commands using the WAIT command, which is written as follows:

   WAIT,<# milliseconds>

8.5.2 Register Editing

The next section describes the four available modes of register editing: *Register Map Editor*, *Encoder Module Editor*, *Generic I2C Register Editor*, and *Property Sheets*. Each of these functions can be selected from the *Edit* menu.

8.5.2.1 Register Map Editor

The register map editor, as shown in Figure 13, allows the display and editing of the entire used register space of the device within a simple scrolling text box. To open this, click on *Edit Register Map* in the *Edit* menu and click on the device type to edit. If the intended device type is not shown, then use the Windows menu to activate the existing window.

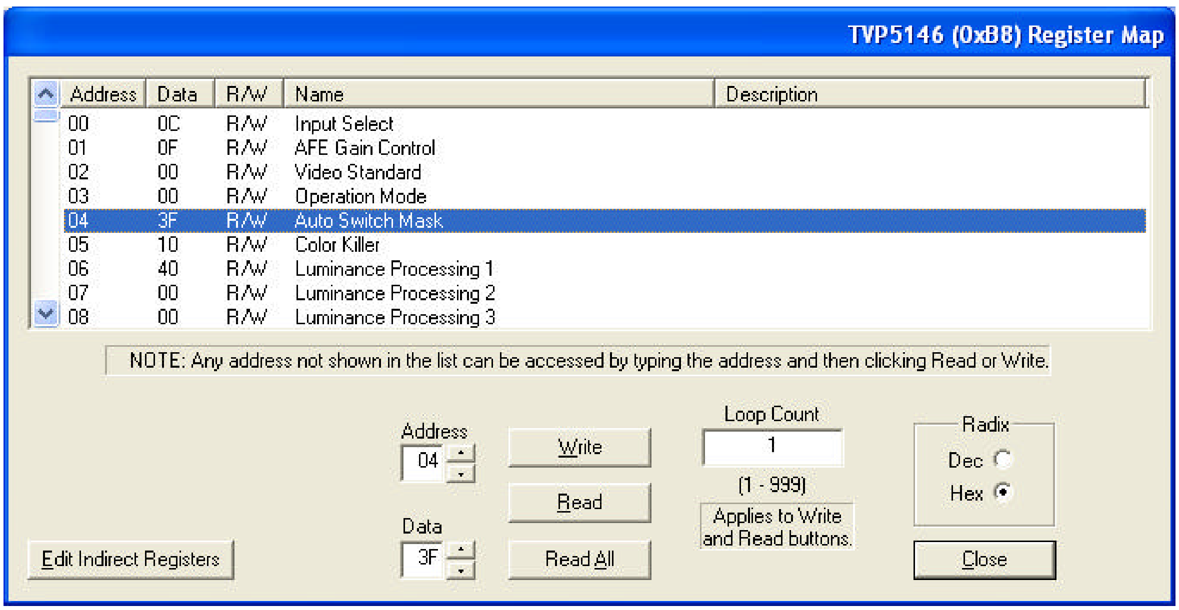


Figure 13. Register Map Editor

Table 5. Register Map Editor Controls

Control	Definition
Register Window	Scrolling text box that displays the address and data for the I^2C registers that are defined for the device.
Address Edit Box	This contains the I^2C subaddress that will be accessed using the *Write* and *Read* buttons. Clicking on a row selects an address, which then appears in the address edit box. NOTE: After clicking on a row, the *Data Edit* box contains the data that was in the register window. The device has not yet been read. The address up/down arrows are used to jump to the next/previous subaddress that is defined for the device. If an address is not defined for the device, then it can still be accessed by typing the subaddress in the *Address Edit* box.
Data Edit Box	This contains the data which will be written to or was read from the I^2C subaddress. The data up/down arrows incr/decr the data value by 1.
Write Button	Writes the byte in the *Data Edit* box to the address in the *Address Edit* box. The I^2C register is written to whether or not the data is different from the last time the register was read.
Read Button	Reads the data from the address in the *Address Edit* box into the *Data Edit* box and the register window.
Read All Button	Reads all defined readable registers from the device and updates the register window.
Hex Button	Converts all values in the register window and address and data edit boxes to hexadecimal.
Dec Button	Converts all values in the register window and address and data edit boxes to decimal.
Close Button	Closes the dialog. NOTE: Multiple edit register map windows can be open at the same time (one for each device). Use the Window menu to navigate.
Loop Count	Causes subsequent write or read operations to be performed N times. N is entered as a decimal number from 1 to 999.
Edit Indirect Registers	Opens the indirect register editor of the TVP5160.

8.5.2.2 Encoder Module Editor

The encoder module editor, as shown in Figure 14, allows the display and editing of the video encoder registers. This editor works like the *Register Map Editor*.

To open this, click on *Edit Register Map* in the *Edit* menu and click on *Encoder Module*.

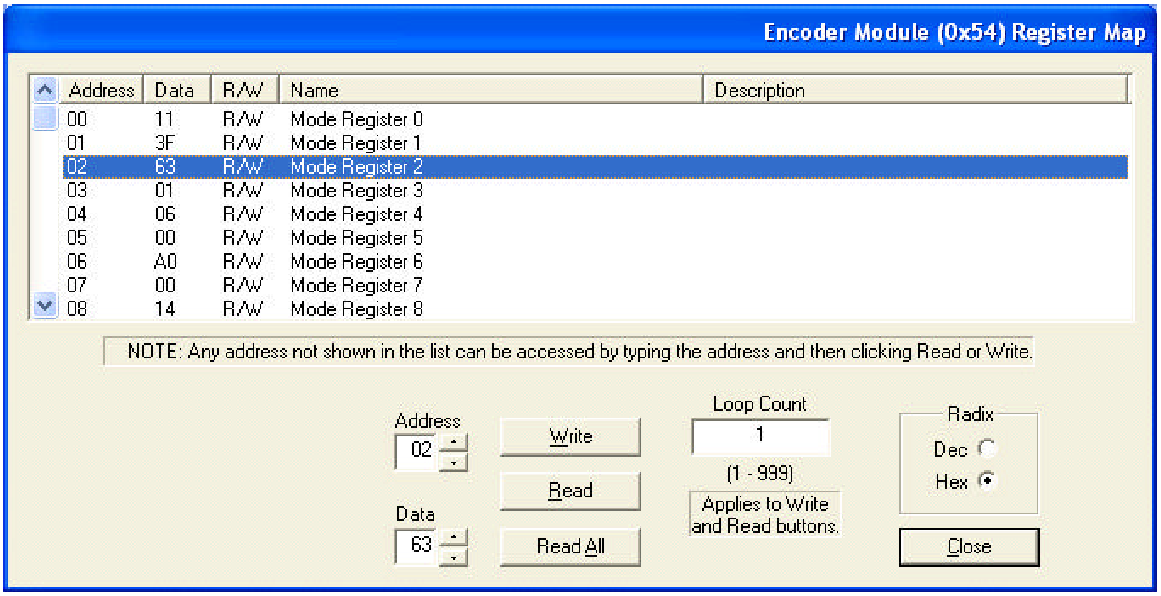


Figure 14. Encoder Module Editor

8.5.2.3 Generic I²C Register Editor

The *Generic I2C Editor*, as shown in Figure 15, allows the display and editing of any device on the I²C bus. This editor works like the *Register Map Editor*, except that the I²C slave address must be entered and the *Read All* button is disabled.

To open this, click on *Edit Register Map* in the *Edit* menu and then click on *Generic I2C*.

The video encoder module registers can be edited using I²C subaddress 0x54 (default) or 0x56, if the alternate slave address is being used.

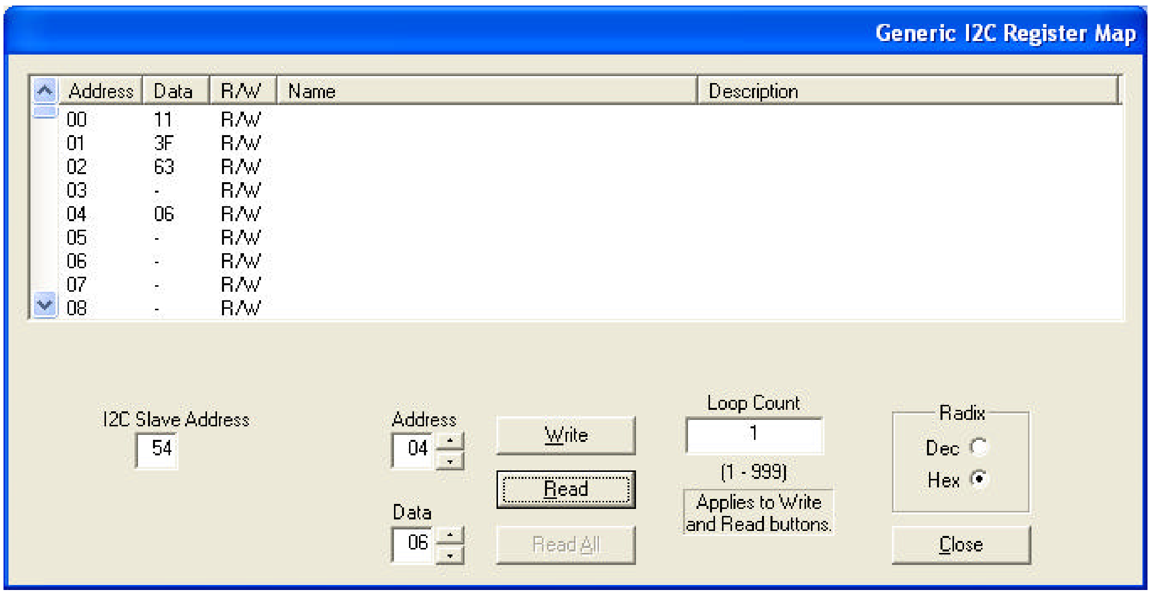


Figure 15. Generic I²C Register Editor

8.5.2.4 Indirect Register Editor

The indirect register editor, as shown in Figure 16, allows the display and editing of the indirect registers (or hardware registers) of the device.

To open this, first open the register map editor in the *Edit* menu for the TVP5160. Then, click the *Edit Indirect Registers* button. The operation of the controls of the indirect register editor is explained in Table 6.

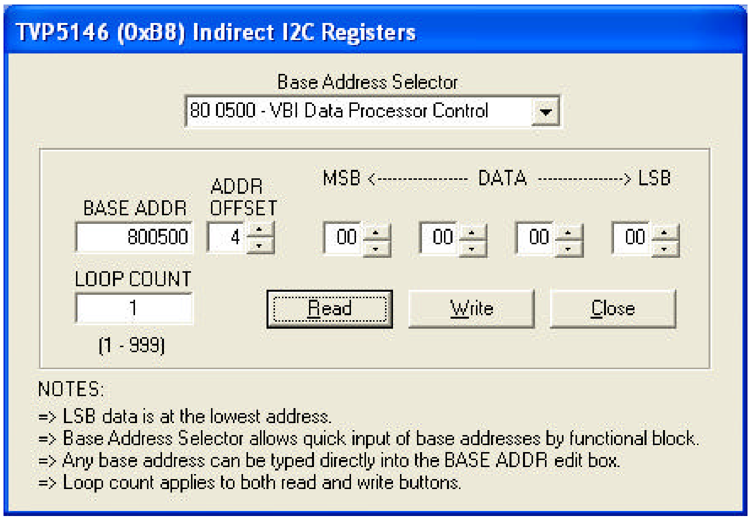


Figure 16. Indirect Register Editor

Table 6. Indirect Register Controls

Control	Definition
Base Address Selector	The indirect registers use a 24-bit address internally. The base address selector allows quick entry of the base address. The list contains base addresses for the major functional blocks of the TVP5160 decoder.
Base Address Edit Box	This allows the 24-bit base address to be typed in directly.
Address Offset Edit Box	This contains the lowest byte of the 24-bit internal address. The full 24-bit address is formed by adding the base address to the address offset. The address up/down arrows increment/decrement the address offset by 4.
Data Edit Boxes	This contains the 32-bit data word that will be written to or read from the indirect address. The LSB data is at the lowest address. The data is written/read LSB first. The data up/down arrows increment/decrement the data value by 1.
Write Button	Writes the (4) bytes in the *Data Edit* boxes starting at the 24-bit indirect address BASE+OFFSET.
Read Button	Reads (4) consecutive data bytes starting at the 24-bit indirect address BASE+OFFSET to the *Data Edit* boxes.
Loop Count Edit Box	Causes subsequent write or read operations to be performed N times. N is entered as a decimal number from 1 to 999.
Close Button	Closes the dialog. NOTE: The indirect register editor can remain open with other windows. Use the Window menu to navigate.

8.5.3 Property Sheets

The *Property Sheets* represent the register data in a user-friendly format. The data is organized by function, with each function having its own page and being selectable via tabs at the top.

To open this, click on *Edit Property Sheets* in the *Edit* menu and select the device type to edit.

When the property sheet function is started or whenever you tab to a different page, all readable registers in the device are read from hardware to initialize the dialog pages. Values on the page are changed by manipulating the various dialog controls as seen in Figure 17.

There are *OK*, *Cancel*, and *Apply* buttons at the bottom of each property page. These are explained in detail below.

8.5.3.1 Reading the Register Map

The property sheets were designed so that the data displayed is always current. Certain actions cause the entire register map to be read from the device and to update the property sheets. This happens when:

1. Property sheets are initially opened.
2. When tabbing from one page to another.
3. When *Read All* is clicked.
4. When making the *Property Sheets* window the active window (by clicking on it).
5. When making a *Register Map Editor* window the active window (by clicking on it).

8.5.3.2 Auto–Update from Device

Items 4 and 5 above are referred to as the *Auto–Update* feature. *Auto–Update* can be disabled by setting its program option button to DISABLED. This button is located on the initial dialog box (WinVCC4 Configuration).

With *Auto–Update* enabled (default), the user can open both the *Property Sheets* and the *Register Map Editor* at the same time. Changes made to the *Property Sheets* (and applied) are updated in the register map window as soon as the *Register Map* window is clicked on. It also works the other way; changes made in the *Register Map Editor* are updated in the *Property Sheets* as soon as the *Property Sheets* window is clicked on.

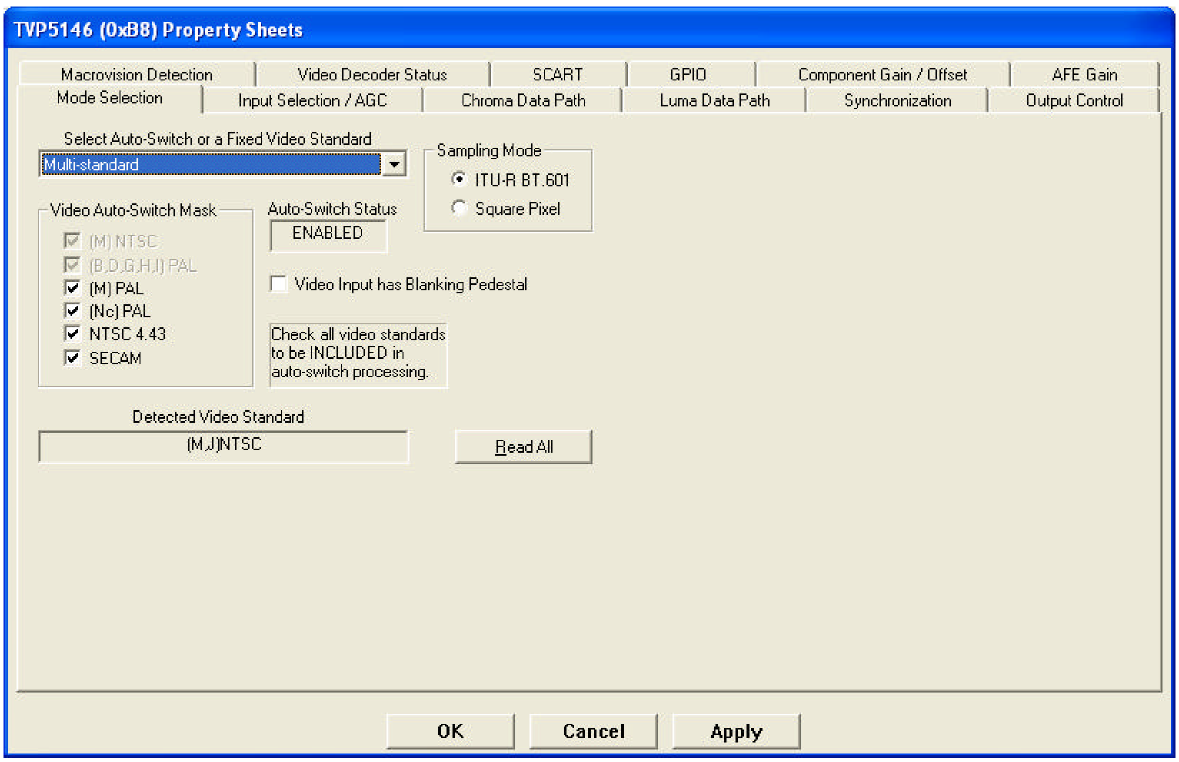


Figure 17. Property Sheets

Table 7. Use of Property Sheet Controls

Dialog Control	What Do I Do With It?	When is Hardware Updated?
Read–Only Edit Box	Read status information	N/A
Check Box	Toggle a single bit	After *Apply*
Drop–Down List	Select from a text list	After *Apply*
Edit Box	Type a number	After *Apply*
Edit Box with Up/Down arrows	Use up/down arrows or type a number	Up/Down Arrows: Immediately Type a number: After *Apply*
Slider	Slide a lever	Immediately
Pushbutton	Initiate an action	Immediately

Table 8. Property Sheet Button Controls

Button Control	Definition
OK	Writes to all writeable registers whose data has changed. A register is flagged as changed if the value to be written is different from the value last read from that address. Closes the dialog.
Cancel	Causes all changes made to the property page since the last *Apply* to be discarded. Changes made to dialog controls with ‘immediate hardware update’ are not discarded, since they have already been changed in hardware. Does not write to hardware. Closes the dialog.
Apply	Writes to all writeable registers whose data has changed. A register is flagged as changed if the value to be written is different from the value last read from that address.

9 Troubleshooting

This chapter discusses ways to troubleshoot the TVP5160EVM.

9.1 Troubleshooting Guide

If you are experiencing problems with the TVP5160EVM hardware or the WinVCC4 software, see Table 9 for available solutions.

Table 9. TVP5160EVM Troubleshooting

Symptom	Cause	Solution
At startup, the error message *Cannot find DLL file DLPORTIO.DLL* appears.	The parallel port driver supplied with the EVM has not been installed.	Run Port95NT.EXE on the CD to install the driver.
Blank screen	Wrong video input is selected.	Go to *Edit->Property Sheets-> TVP5160*, Analog Video page, select the correct video input(s) and click *Apply*. (The Composite Video 1 input is default.)
	Source is connected to the wrong input connector.	Connect source to the correct input connector.
	YCbCr outputs or clock output is disabled.	Go to *Edit->Property Sheets-> TVP5160*, Output Control, check the *Enable YCbCr Outputs* and *Enable Clock Outputs* check boxes and click *Apply*.
No color	GLCO pin is not set to output the GLCO signal.	Go to *Edit->Property Sheets-> TVP5160*, GPIO page, set the drop down box labeled “GLCO/I2CA Terminal” to Genlock Output and click *Apply*.
Screen colors are only magenta and green.	Wrong YCbCr output format.	Go to *Edit->Property Sheets-> TVP5160*, Output Control, set the YCbCr output format to 10-bit 4:2:2 ITU-R BT.656 with embedded syncs mode and click *Apply*.
Video standard auto–switch does not work on the video decoder side.	Autoswitch masks are not set correctly.	Go to *Edit->Property Sheets-> TVP5160*, Mode Selection page, check all standards to be included in auto–switch processing and click *Apply*.
	Video decoder is not in auto–switch mode.	Go to *Edit->Property Sheets-> TVP5160*, Mode Selection page, set the drop–down box to *Multi–Standard* and click *Apply*.
Video standard auto–switch does not work on the video encoder side.	Auto–switch polling is not enabled.	Click *Real–Time Polling* in the Tools menu. Click *Enable All* and *OK*.

Table 10. I^2C Troubleshooting

Symptom	Cause	Solution
No I^2C communication	I^2C slave address is wrong.	Close and restart WinVCC4. Choose the alternate slave address in the WinVCC4 Configuration dialog.
	Parallel cable is not connected from PC parallel port to the EVM DB25 connector.	Connect cable.
	EVM is not powered on.	The power supply must be plugged into a 100-V to 240-V/ 47-Hz to 63-Hz power source and the cord must be plugged into the power connector on the EVM.
	Wrong type of parallel cable.	Some parallel cables are not wired straight through pin-for-pin. Use the cable supplied with the EVM.
	PC parallel port mode is not set correctly.	Reboot PC, enter BIOS setup program, set parallel port LPT1 mode (Addr 378h) to ECP mode or bidirectional mode (sometimes called PS/2 mode or byte mode). If already set to one of these two modes, switch to the other setting. See Section 4.2.1.
	Device was placed in power–down mode.	Press the reset button on the TVP5160EVM.
	EVM was configured for an external I^2C master.	Reinstall 0-? resistors R5 and R6. Control EVM using the PC parallel port.
	Still no I^2C communication	The PC may not be capable of operating in the required parallel port mode. This is true of some laptop computers. Use a different computer, preferably a desktop PC.

When WinVCC4 is started and the WinVCC4 Configuration dialog box is closed with *OK*, the I^2C system test is performed (unless the I2C System Test program options button was disabled).

If the I^2C system test fails, a dialog box will appear. Figure 18 reports that a read from TVP5160 failed, using slave address 0xB8, subaddress 0xBD. The data read was 0x00. The expected data was 0x01.

After noting which device had a problem, click *OK* to continue. Next, the *Corrective Action Dialog* box appears to help fix the problem.

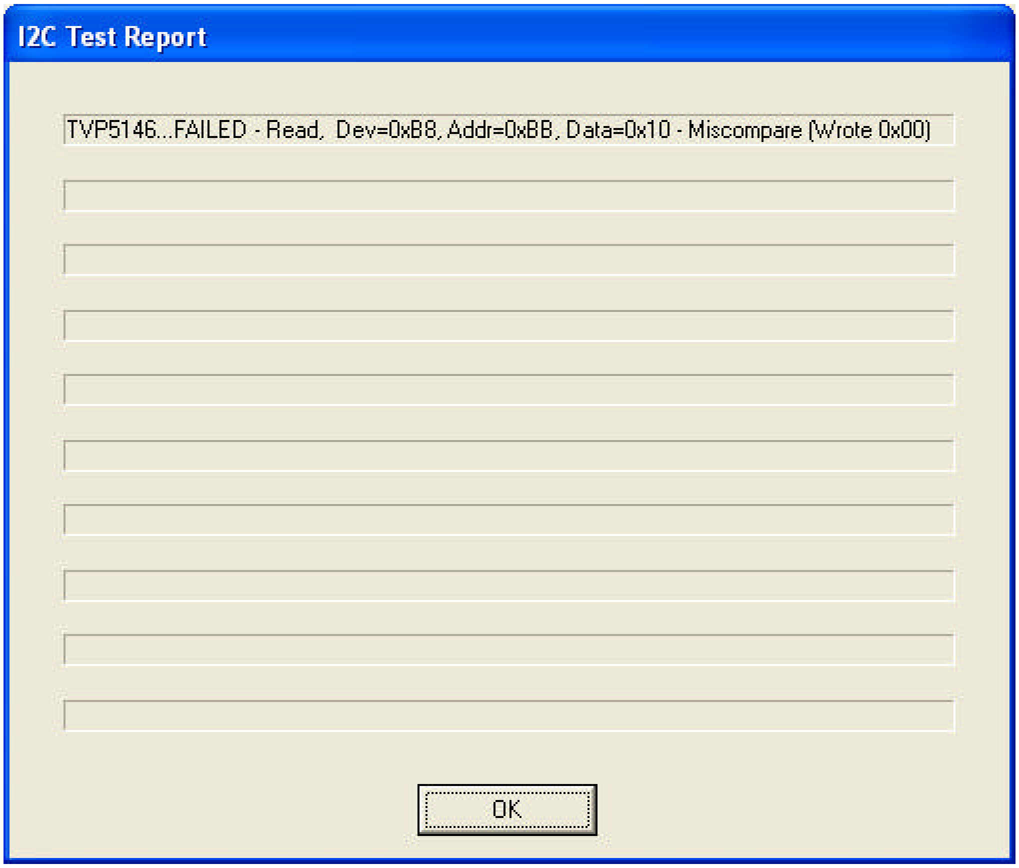


Figure 18. I^2C System Failure Dialog Box

9.2 Corrective Action Dialogs

After closing the I^2C system test report dialog box, the dialog in Figure 19 appears.



Figure 19. Corrective Action Dialog Box

1. If the cable is NOT connected from the PC parallel port to the TVP5160EVM or if the EVM power is not on:
   a. Click *NO*.
   b. The dialog shown in Figure 20 appears instructing you to correct the problem.

c. Correct the problem.

d. Click *OK* to continue. The real-time polling dialog should appear. See Section 3.2.3.

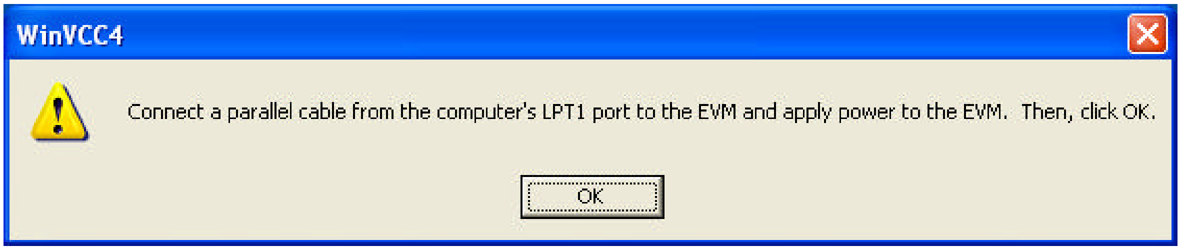


Figure 20. Corrective Action Required

2. If the cable is connected from the PC parallel port to the TVP5160EVM AND the EVM power is on:

   a. Click *Yes*.

   b. The dialog shown in Figure 21 appears. This dialog appears if the PC parallel port mode setting may need to be changed.

   NOTE: Only run the PC BIOS setup program if the I²C communication problem cannot be resolved in another way. (Correct slave address settings, reset or power cycle the EVM and/or check that the device type selected was TVP5160).

   c. Click *OK* to continue.

   d. The real-time polling dialog appears. Click *OK* to close it and get to the main menu.

   e. Click *Exit* in the *File* menu to exit the program.

   f. See troubleshooting guide above.

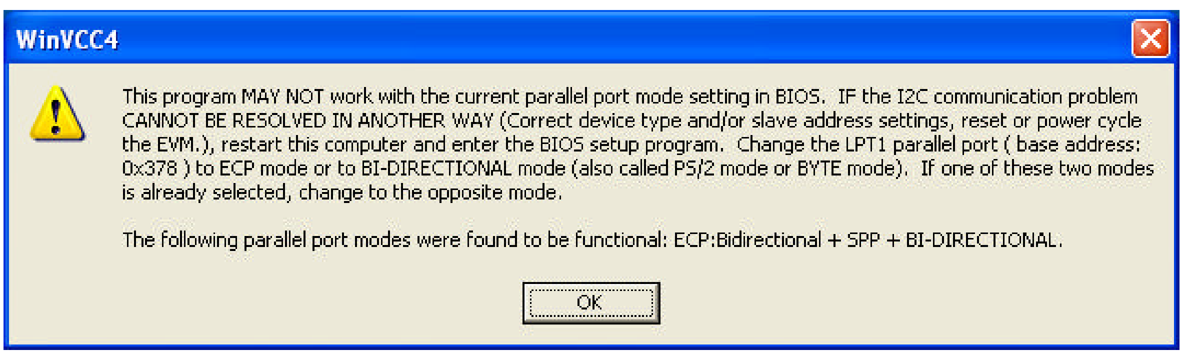


Figure 21. Corrective Action Required

9.2.1 Setting the PC Parallel Port Mode

NOTE: Only run the PC BIOS setup program if the I²C communication problem cannot be resolved in another way. (Correct slave address settings, reset or power cycle the EVM, and/or check that the device type selected was TVP5160).

1. Restart the PC.
2. During the boot process, enter the BIOS setup program by pressing the required key (usually the initial text screen tells you which key to press).
3. Find the place where the parallel port settings are made.
4. Set the parallel port LPT1 at address 378h to ECP mode or bidirectional mode (sometimes called PS/2 mode or byte mode). If one of these two modes is already selected, change to the opposite mode.
5. Exit and save changes.

9.2.2 General I^{2}C Error Report

The error report shown in Figure 22 appears when an I^{2}C error occurs at any time other than after the I^{2}C system test. In this example, there was an acknowledge error at slave address 0x54 (the video decoder module). The error occurred on *Read Cycle Phase 1* on the device (slave) address byte.



Figure 22. I^{2}C Error

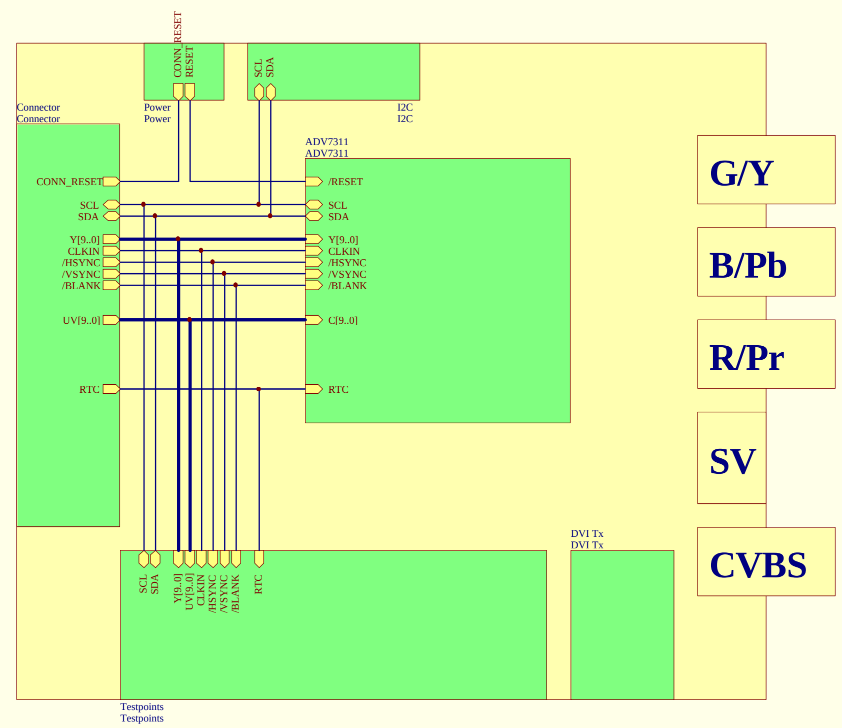
10 TVP5160EVM Schematics

This chapter contains the TVP5160EVM schematics.

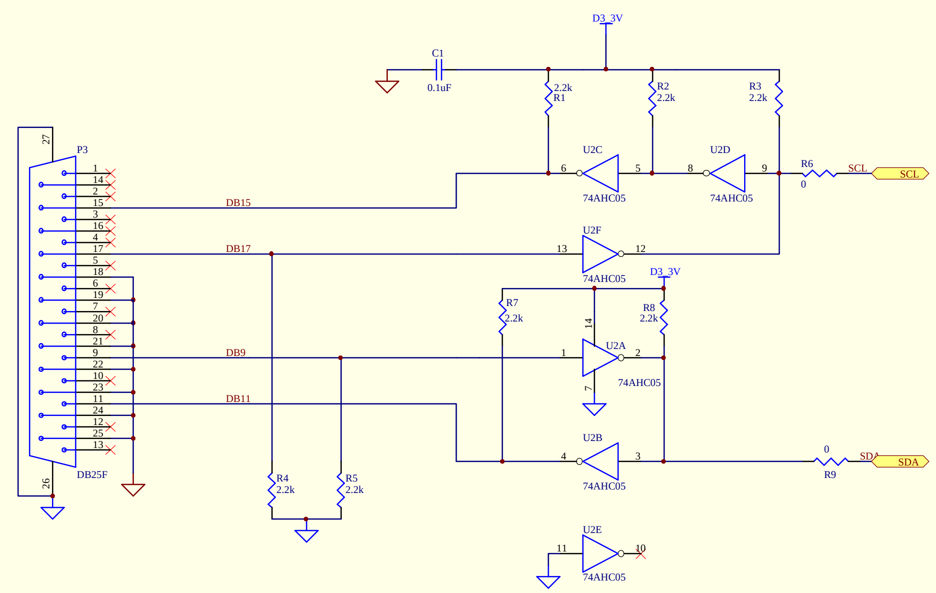
ADV7311 Board
Revision 1.0

Feb 2005

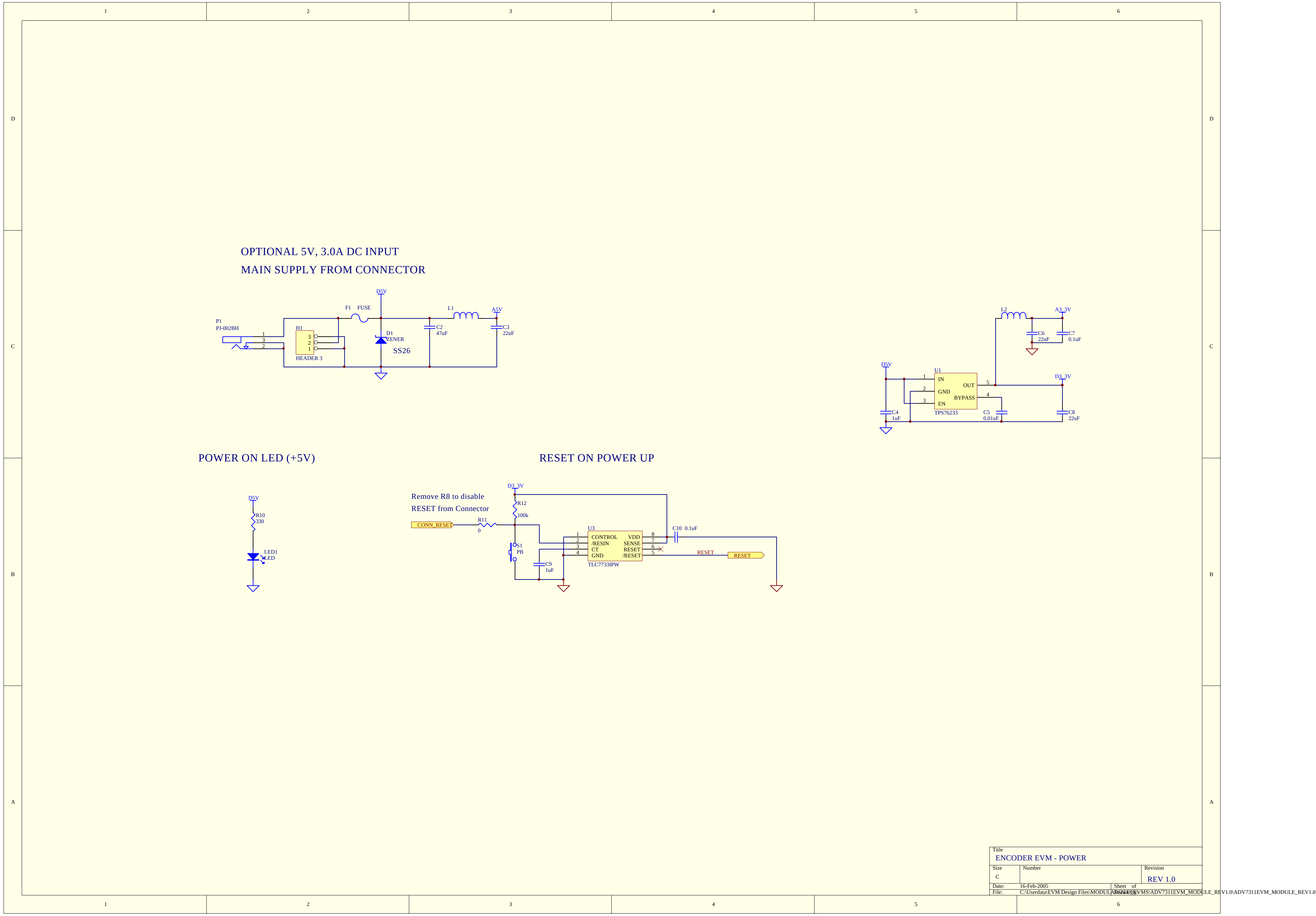
ADV7311 MODULE REV 1.0

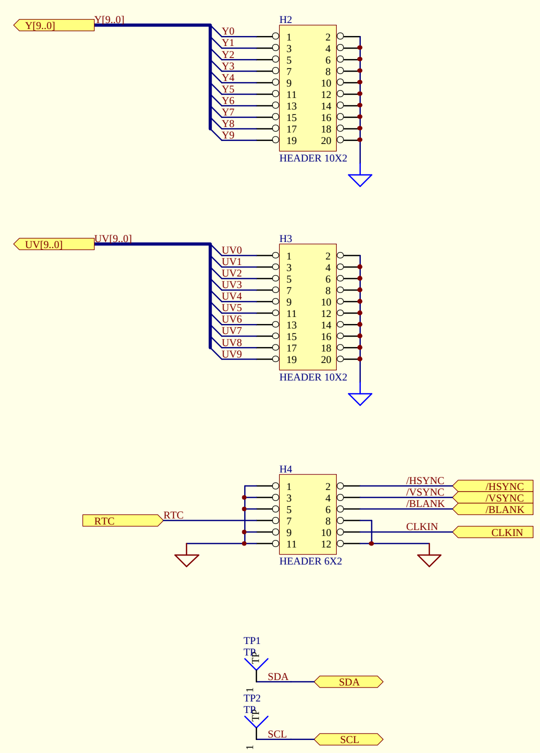


Title		
ENCODER EVM - BLOCK DIAGRAM		
Size: C	Number	Revision: REV 1.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb		

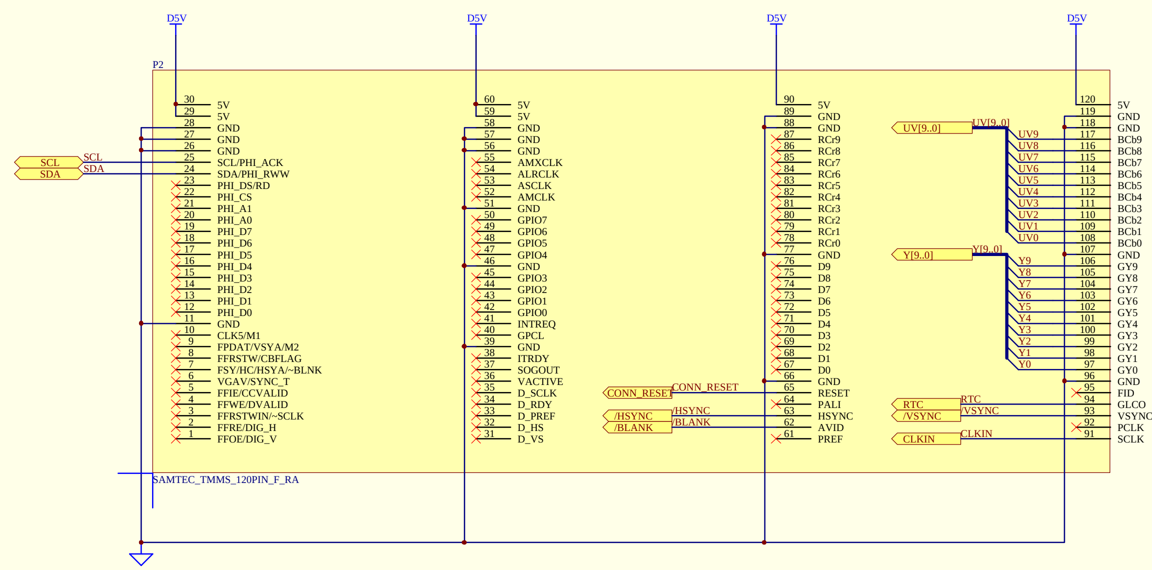


Title		
ENCODER EVM - I2C		
Size: C	Number	Revision: REV 1.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb	Drawn By:	


OPTIONAL 5V, 3.0A DC INPUT
MAIN SUPPLY FROM CONNECTOR
P1
PJ-002BH
H1
HEADER 3
F1 FUSE
D5V
D1
ZENER
SS26
C2
47uF
L1
A5V
C3
22uF
L2
A3_3V
C6
22uF
C7
0.1uF
D5V
U1
IN
GND
EN
OUT
BYPASS
TPS76233
D3_3V
C4
1uF
C5
0.01uF
C8
22uF
POWER ON LED (+5V)
D5V
R10
330
LED1
LED
RESET ON POWER UP
Remove R8 to disable
RESET from Connector
CONN_RESET
R11
0
D3_3V
R12
100k
S1
PB
C9
1uF
U3
CONTROL
/RESIN
CT
GND
VDD
SENSE
RESET
/RESET
TLC7733IPW
C10 0.1uF
RESET
RESET
Title
ENCODER EVM - POWER
Size
C
Number
Revision
REV 1.0
Date: 16-Feb-2005
Sheet of
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb
Drawn By:



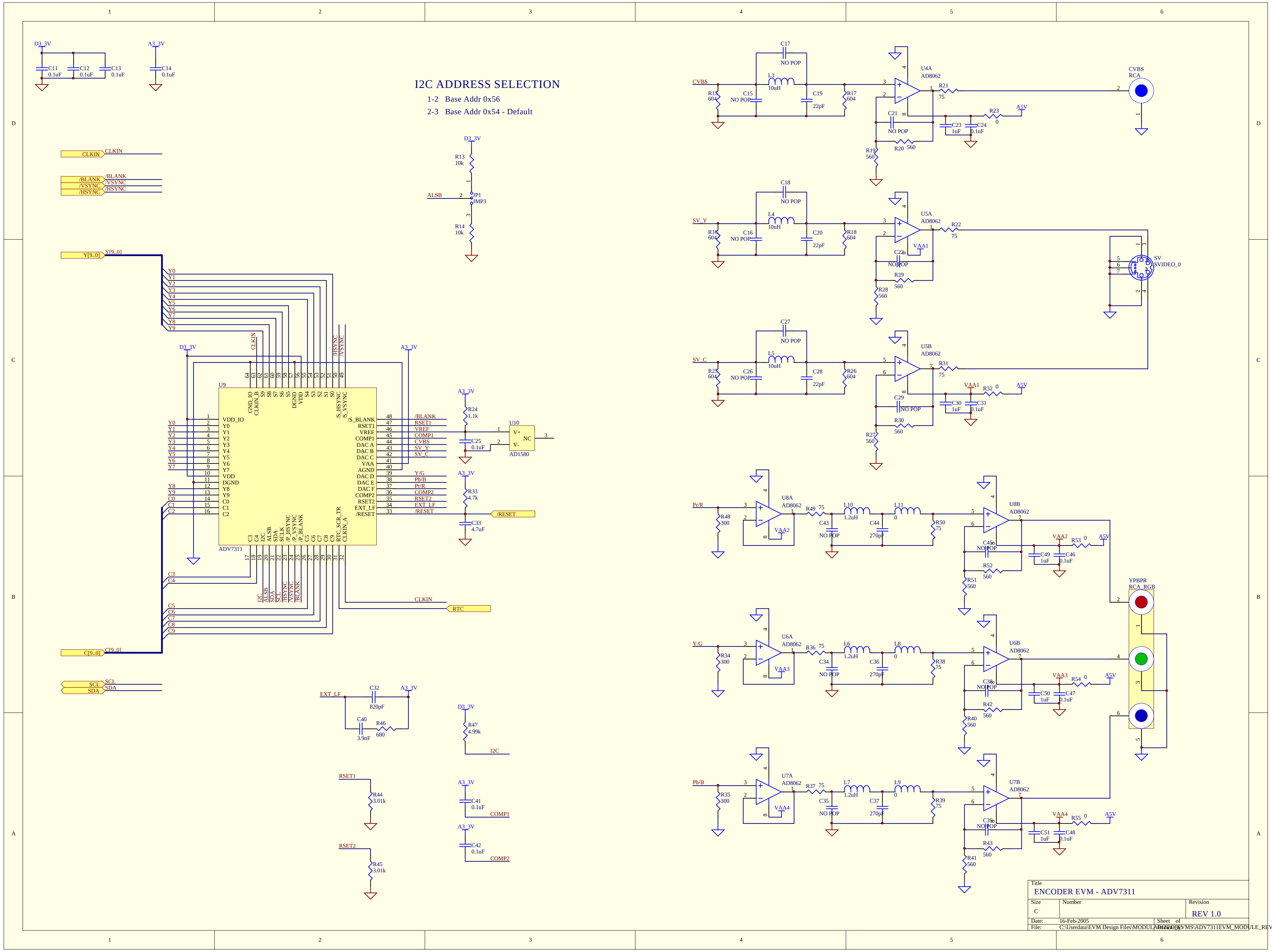
Title	ENCODER EVM - TESTPOINTS		
Size: C	Number		Revision: REV 1.0
Date:	16-Feb-2005	Sheet of	
File:	C:\Userdata\EVM Design Files\MODULARIZED EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb	Drawn By:	

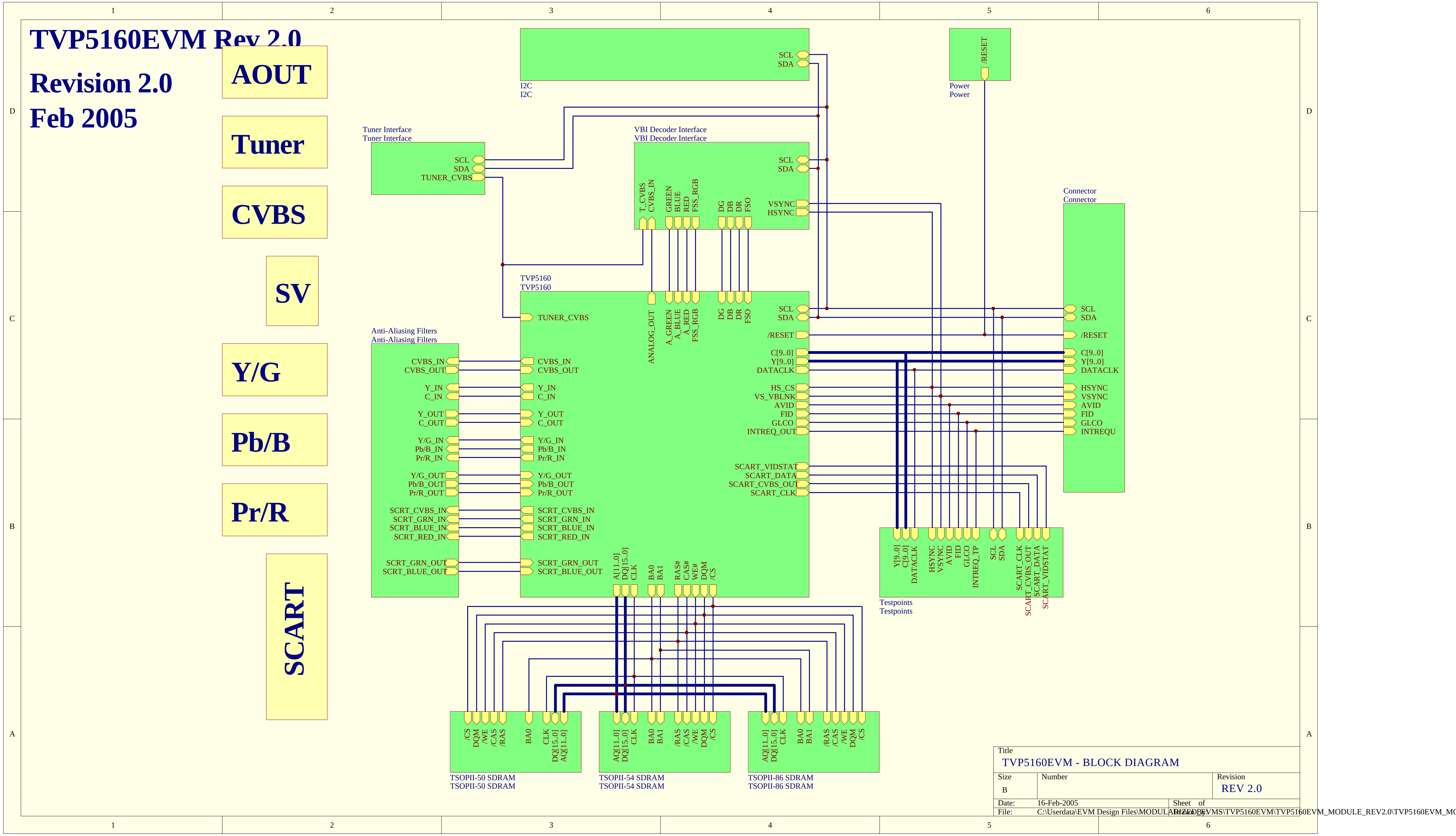


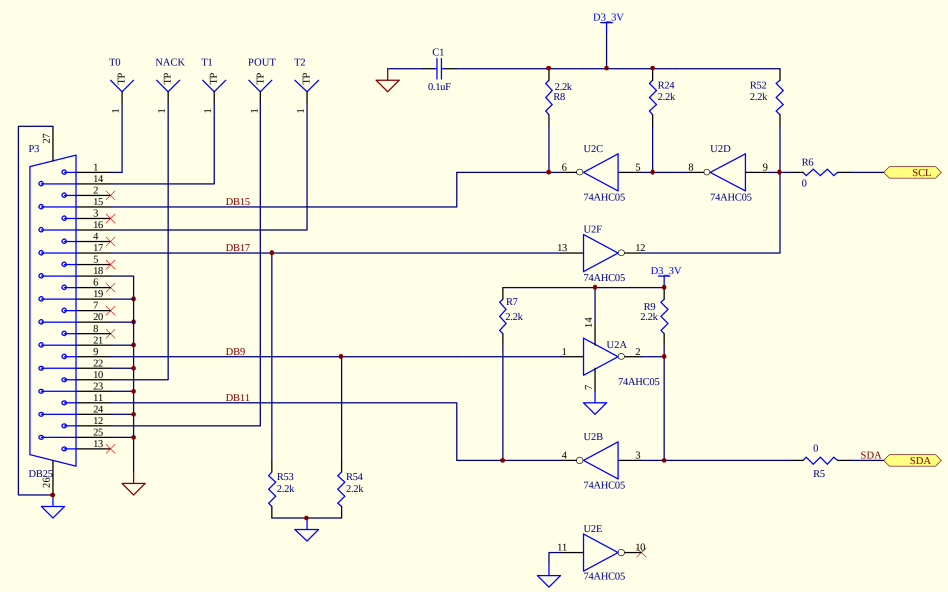
Title
ENCODER EVM - CONNECTOR

Size	Number	Revision
C		REV 1.0

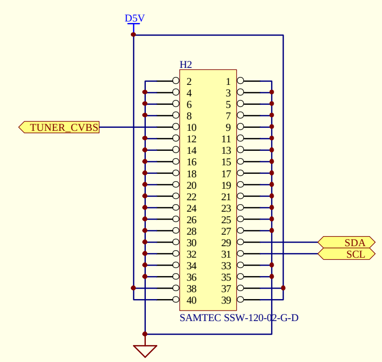
Date: 16-Feb-2005 Sheet of
File: C:\Userdata\EVM Design Files\MODULAR\Encoder EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb


I2C ADDRESS SELECTION
1-2 Base Addr 0x56
2-3 Base Addr 0x54 - Default
D3_3V
A3_3V
CLKIN
/BLANK
/VSYNC
/HSYNC
Y[9..0]
C[9..0]
SCL
SDA
ALSB
JP1
JMP3
R13
10k
R14
10k
U9
ADV7311
U10
AD1580
/RESET
RTC
EXT_LF
RSET1
RSET2
COMP1
COMP2
CVBS
SV_Y
SV_C
Pr/R
Y/G
Pb/B
U4A
AD8062
U5A
AD8062
U5B
AD8062
U8A
AD8062
U8B
AD8062
U6A
AD8062
U6B
AD8062
U7A
AD8062
U7B
AD8062
CVBS
RCA
SV
SVIDEO_0
YPBPR
RCA_RGB
A5V
VAA1
VAA2
VAA3
VAA4
NO POP
ENCODER EVM - ADV7311
Size
C
Number
Revision
REV 1.0
Date: 16-Feb-2005
File: C:\Userdata\EVM Design Files\MODULAR BOARD EVMS\ADV7311EVM_MODULE_REV1.0\ADV7311EVM_MODULE_REV1.0.Ddb

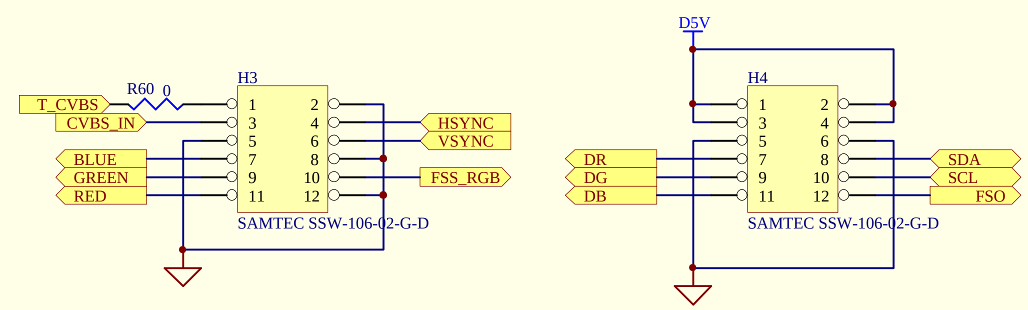

TVP5160EVM Rev 2.0
Revision 2.0
Feb 2005
AOUT
Tuner
CVBS
SV
Y/G
Pb/B
Pr/R
SCART
I2C
SCL
SDA
Power
/RESET
Tuner Interface
TUNER_CVBS
VBI Decoder Interface
T_CVBS
CVBS_IN
GREEN
BLUE
RED
FSS_RGB
DG
DB
DR
FSO
VSYNC
HSYNC
TVP5160
ANALOG_OUT
A_GREEN
A_BLUE
A_RED
C[9..0]
Y[9..0]
DATACLK
HS_CS
VS_VBLNK
AVID
FID
GLCO
INTREQ_OUT
SCART_VIDSTAT
SCART_DATA
SCART_CVBS_OUT
SCART_CLK
Anti-Aliasing Filters
CVBS_OUT
Y_IN
C_IN
Y_OUT
C_OUT
Y/G_IN
Pb/B_IN
Pr/R_IN
Y/G_OUT
Pb/B_OUT
Pr/R_OUT
SCRT_CVBS_IN
SCRT_GRN_IN
SCRT_BLUE_IN
SCRT_RED_IN
SCRT_GRN_OUT
SCRT_BLUE_OUT
A[11..0]
DQ[15..0]
CLK
BA0
BA1
RAS#
CAS#
WE#
DQM
/CS
Connector
INTREQU
Testpoints
INTREQ_TP
/RAS
/CAS
/WE
AQ[11..0]
TSOPII-50 SDRAM
TSOPII-54 SDRAM
TSOPII-86 SDRAM
Title
TVP5160EVM - BLOCK DIAGRAM
Size
B
Number
Revision
REV 2.0
Date: 16-Feb-2005
Sheet of
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO



Title		
TVP5160EVM - I2C		
Size	Number	Revision
C		REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MODULE_REV 2.0.ddb	Drawn By:	



Title		
TVP5146EVM - TV TUNER INTERFACE		
Size: C	Number:	Revision: REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MODULE_REV 2.0.ddb	Drawn By:	

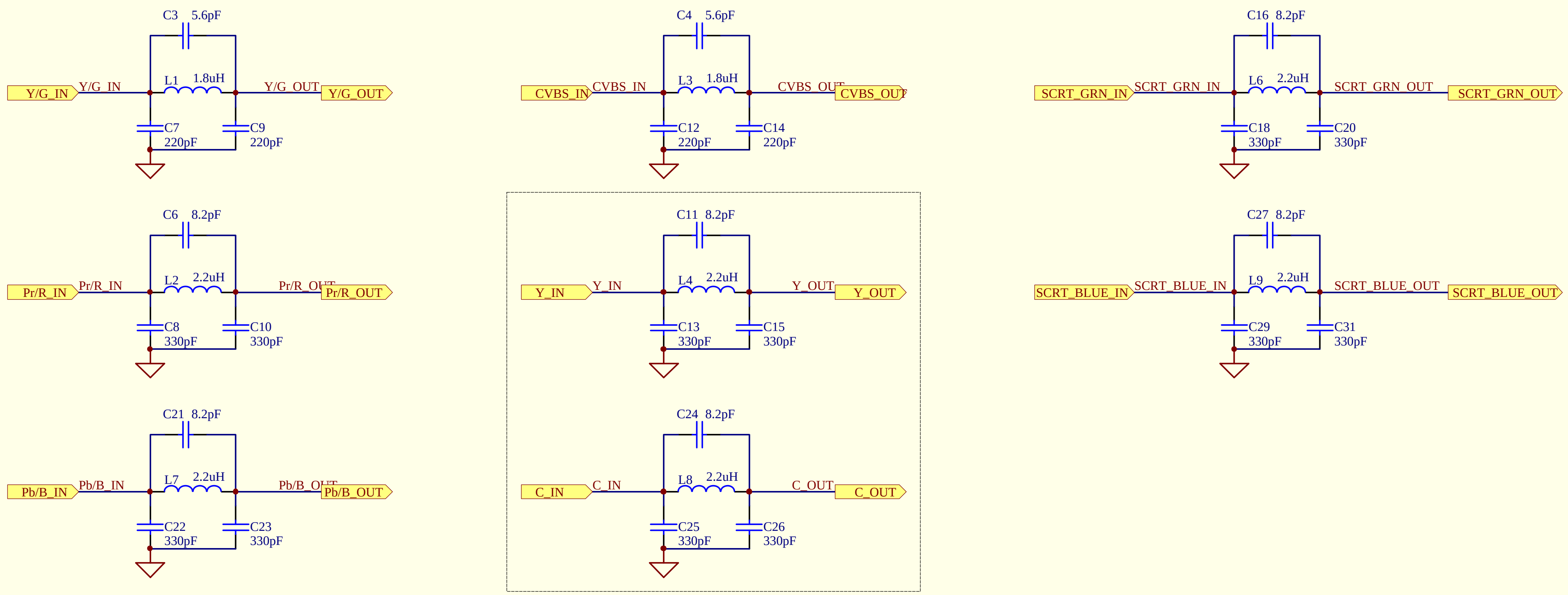


Title		
TVP5160EVM - VBI DECODER INTERFACE		
Size: B	Number	Revision: REV 2.0
Date: 16-Feb-2005		Sheet of
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO		Drawn By:

COMPONENT - YPbPr

CVBS & S-VIDEO

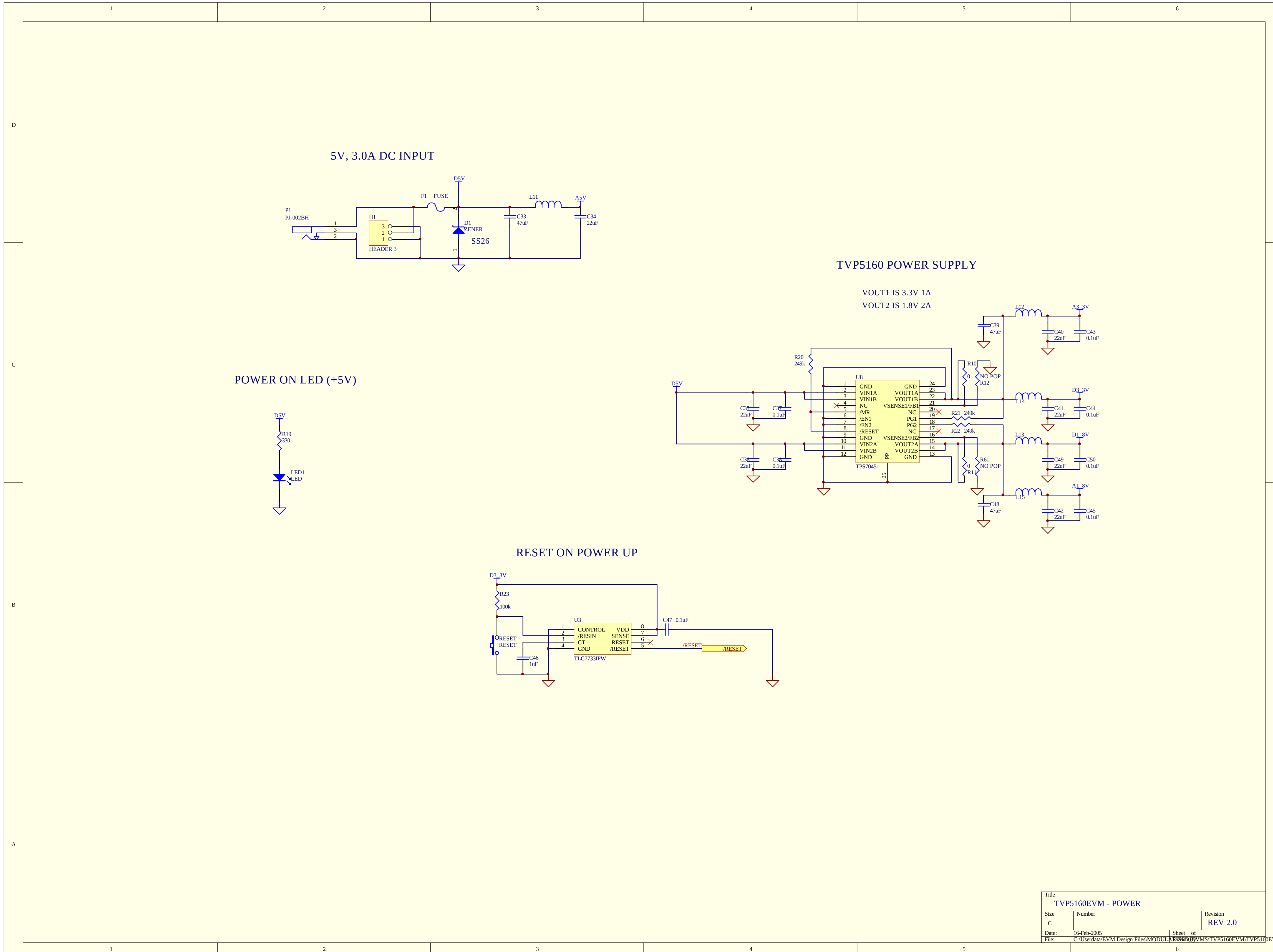
SCART - CVBS & RGB (RED & CVBS shared w/ SV)

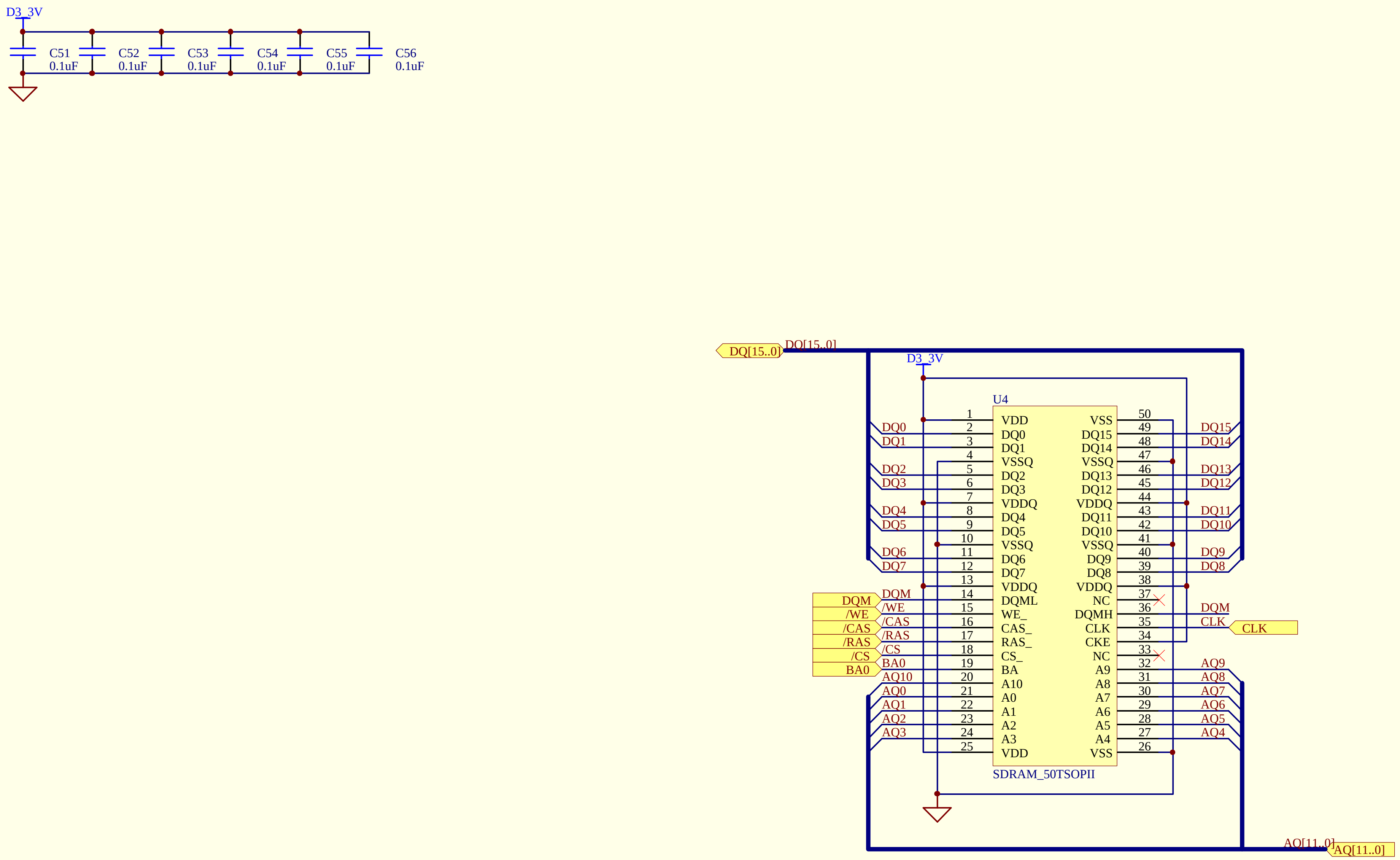


NOTE: SV- (Y/C) is shared with SCART - (CVBS/RED)

These anti-aliasing filters are used for both sets of inputs.

Title: TVP5160EVM - ANTI-ALIASING FILTERS		
Size: B	Number	Revision: REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO	Drawn By:	

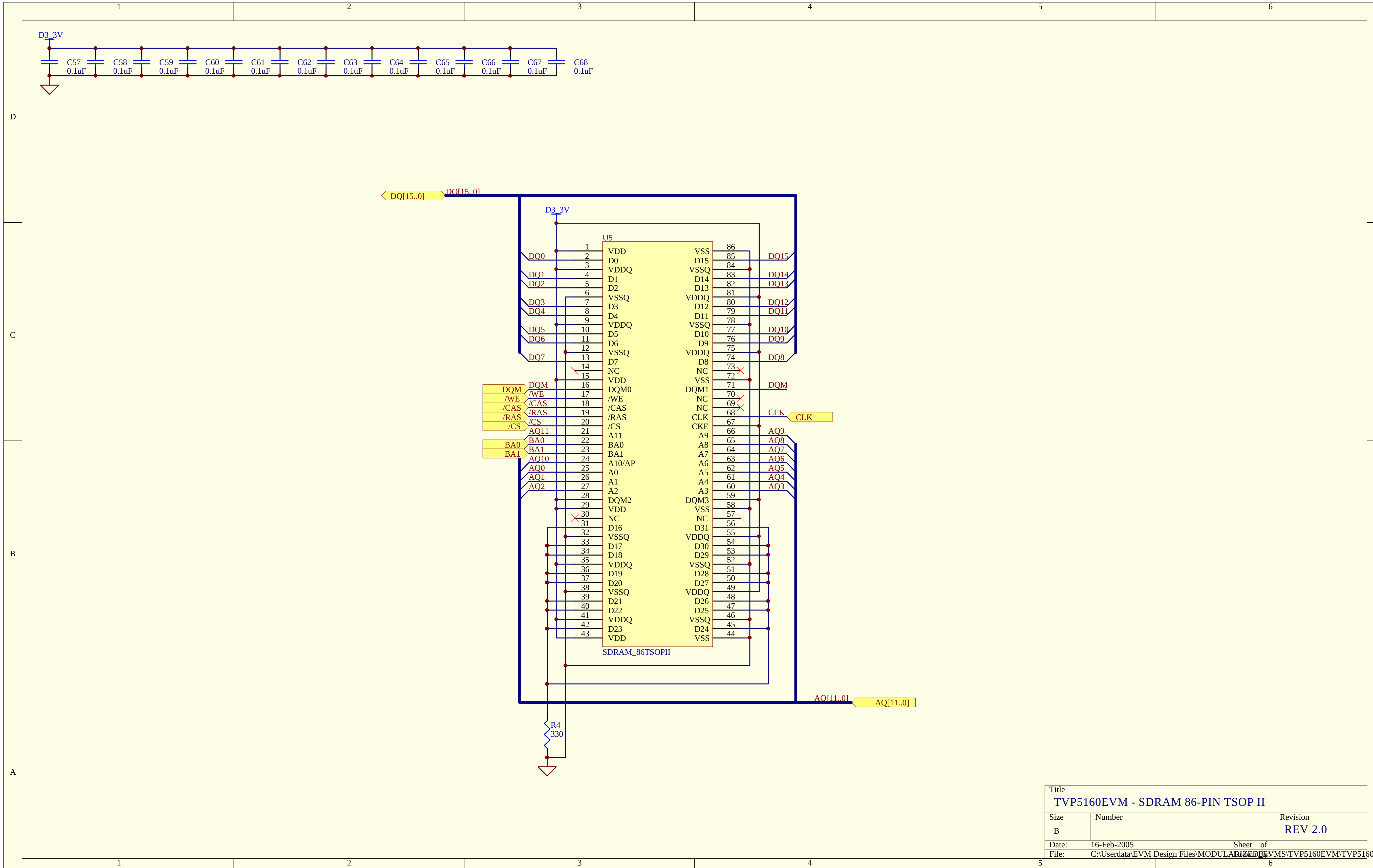

5V, 3.0A DC INPUT
TVP5160 POWER SUPPLY
VOUT1 IS 3.3V 1A
VOUT2 IS 1.8V 2A
POWER ON LED (+5V)
RESET ON POWER UP
P1
PJ-002BH
H1
HEADER 3
F1
FUSE
D1
ZENER
SS26
L11
C33
47uF
C34
22uF
D5V
A5V
U8
TPS70451
R20
249k
R21
249k
R22
249k
NO POP
A3_3V
D3_3V
D1_8V
A1_8V
R19
330
LED1
LED
U3
TLC7733IPW
R23
100k
C46
1uF
C47
0.1uF
/RESET
Title
TVP5160EVM - POWER
Size
C
Number
Revision
REV 2.0
Date:
16-Feb-2005
Sheet of
File:



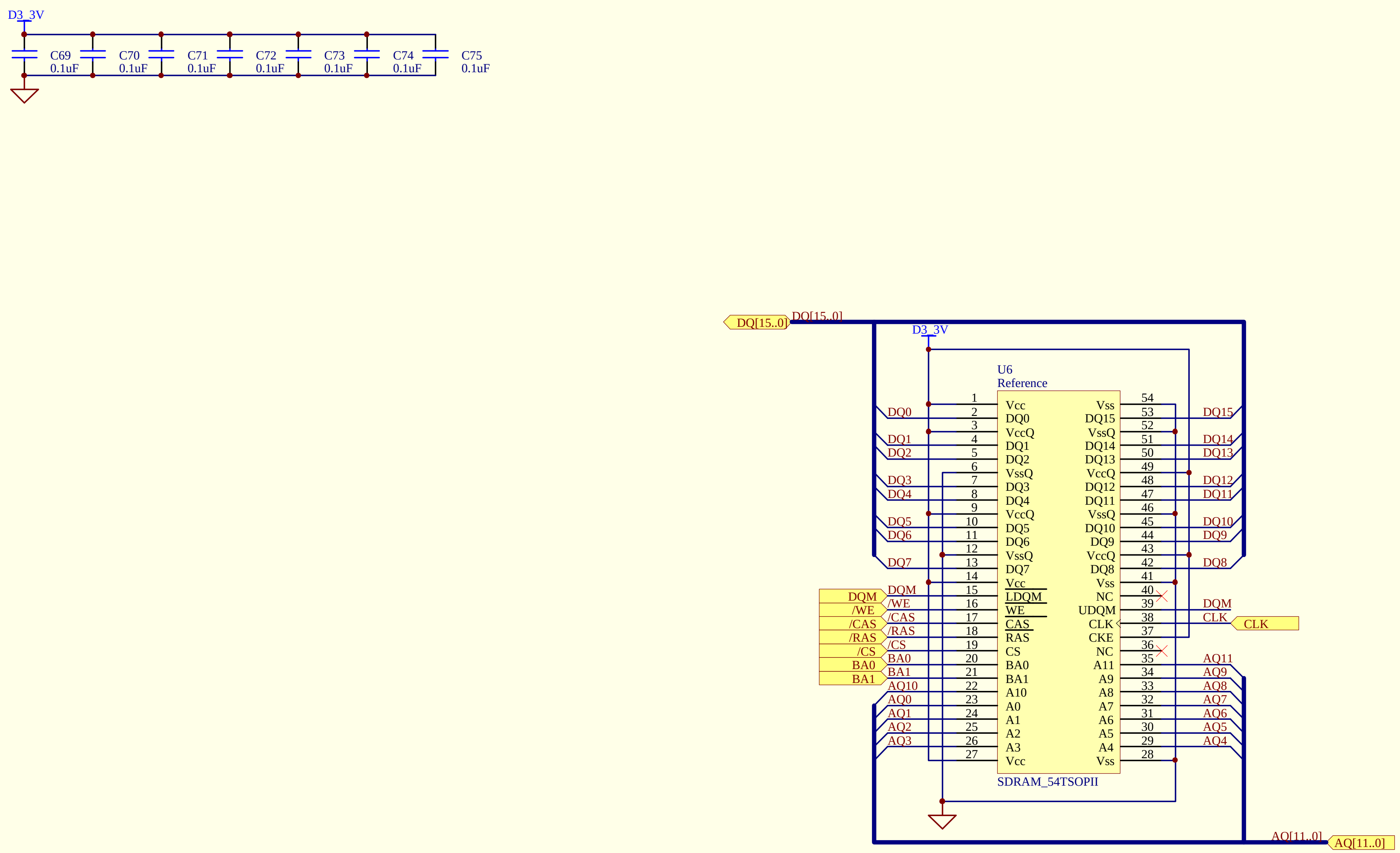
Title
TVP5160EVM - SDRAM 50-PIN TSOP II

Size	Number	Revision
B		REV 2.0

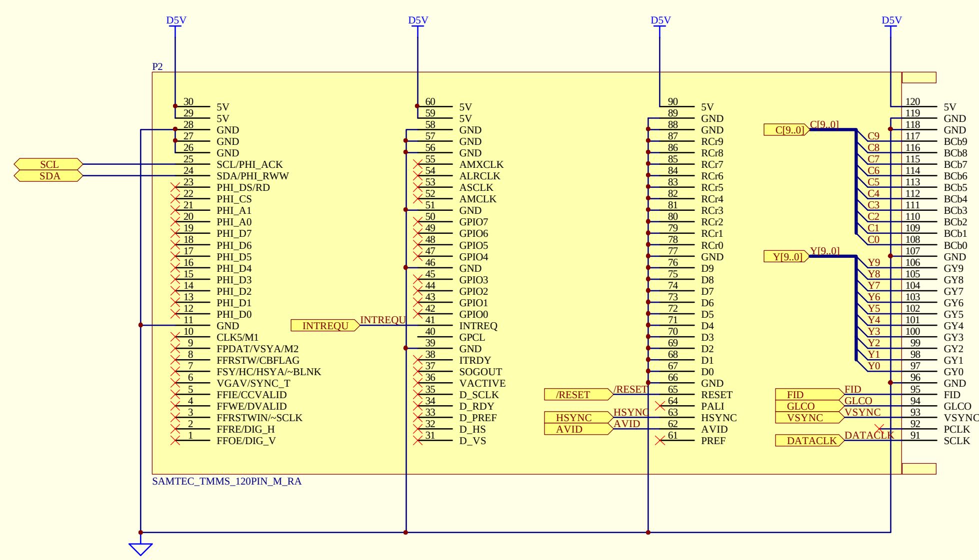
Date: 16-Feb-2005 Sheet of
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO



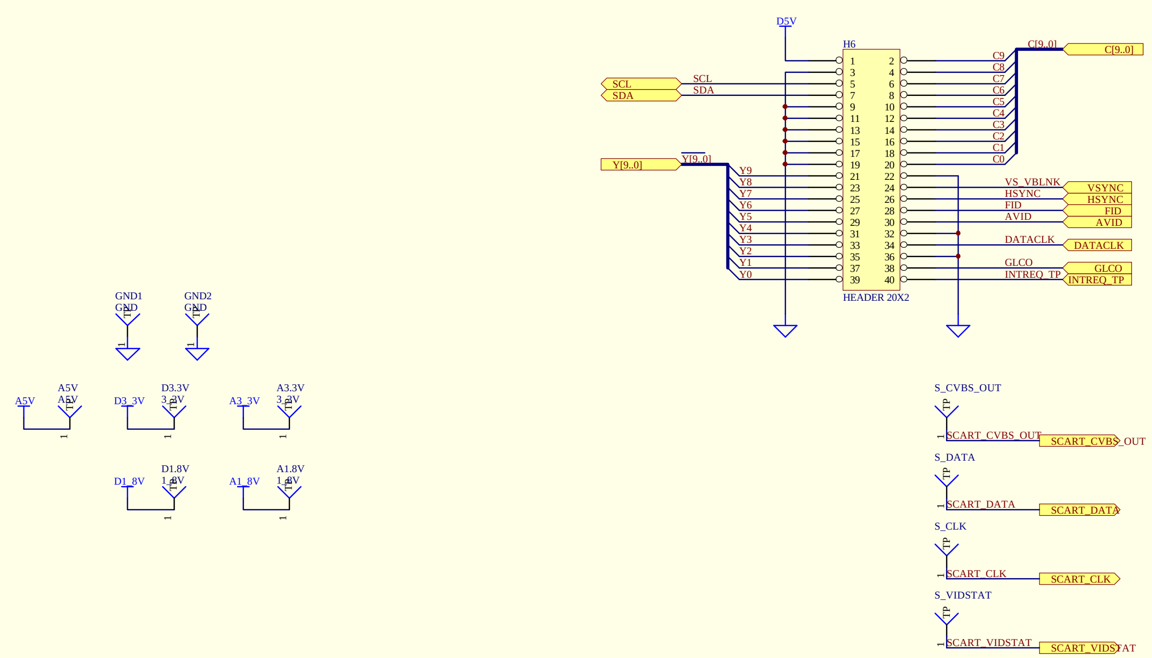
D3_3V
C57 0.1uF
C58 0.1uF
C59 0.1uF
C60 0.1uF
C61 0.1uF
C62 0.1uF
C63 0.1uF
C64 0.1uF
C65 0.1uF
C66 0.1uF
C67 0.1uF
C68 0.1uF
DQ[15..0]
AQ[11..0]
U5
SDRAM_86TSOPII
DQM
/WE
/CAS
/RAS
/CS
BA0
BA1
CLK
R4
330
Title
TVP5160EVM - SDRAM 86-PIN TSOP II
Size
B
Number
Revision
REV 2.0
Date: 16-Feb-2005
Sheet of
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO
Drawn By:



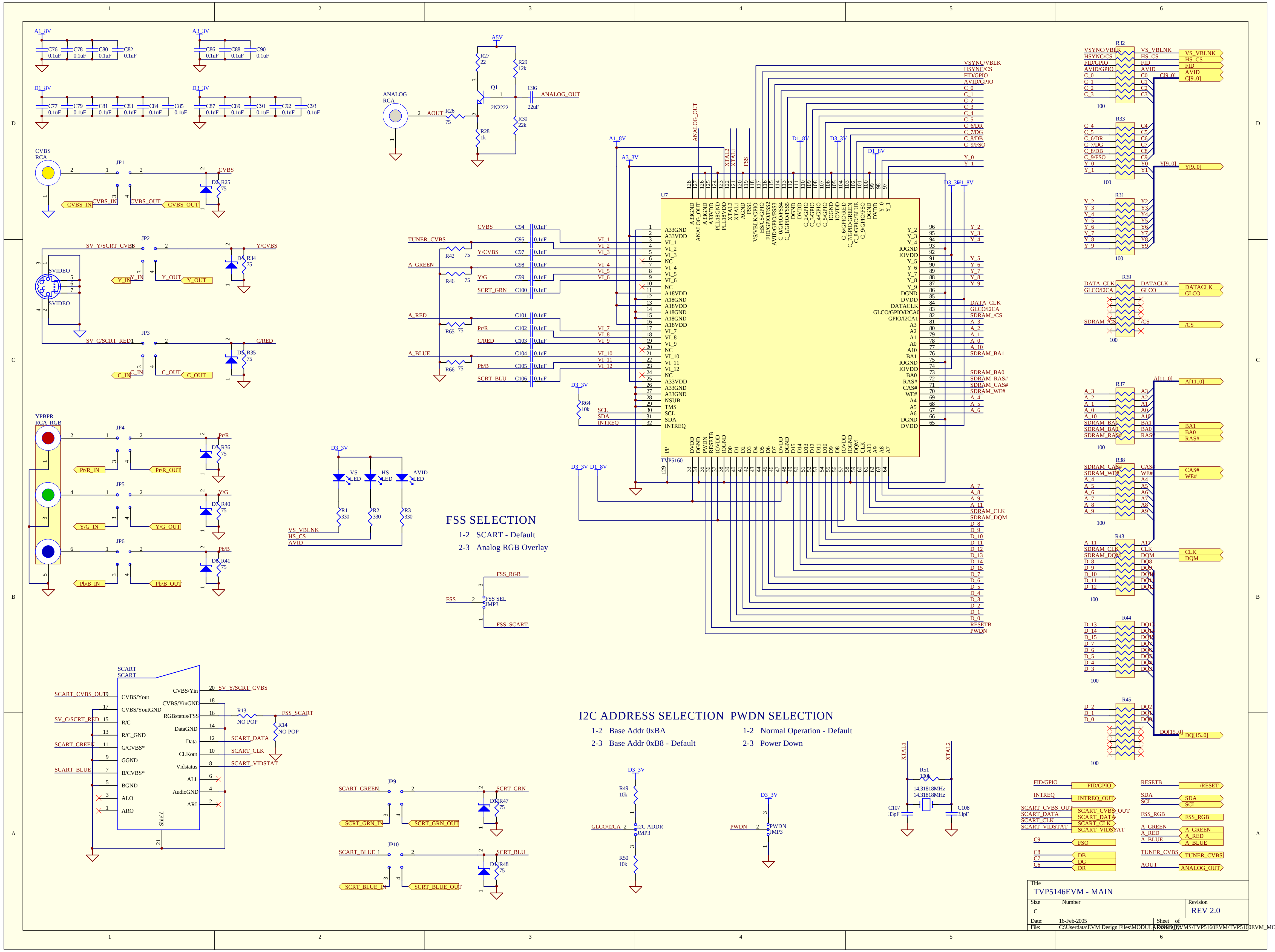
Title		
TVP5160EVM - SDRAM 54-PIN TSOP II		
Size: B	Number	Revision: REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MO	Drawn By:	



Title: TVP5160EVM - CONNECTOR		
Size: C	Number	Revision: REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULAR EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MODULE_REV 2.0.ddb	Drawn By:	



Title: TVP5160EVM - TESTPOINTS		
Size: C	Number:	Revision: REV 2.0
Date: 16-Feb-2005	Sheet of	
File: C:\Userdata\EVM Design Files\MODULARIZED_EVMS\TVP5160EVM\TVP5160EVM_MODULE_REV2.0\TVP5160EVM_MODULE_REV 2.0.ddb	Drawn By:	


FSS SELECTION
1-2 SCART - Default
2-3 Analog RGB Overlay
I2C ADDRESS SELECTION
1-2 Base Addr 0xBA
2-3 Base Addr 0xB8 - Default
PWDN SELECTION
1-2 Normal Operation - Default
2-3 Power Down
TVP5146EVM - MAIN
Size C
Revision REV 2.0
Date: 16-Feb-2005

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265